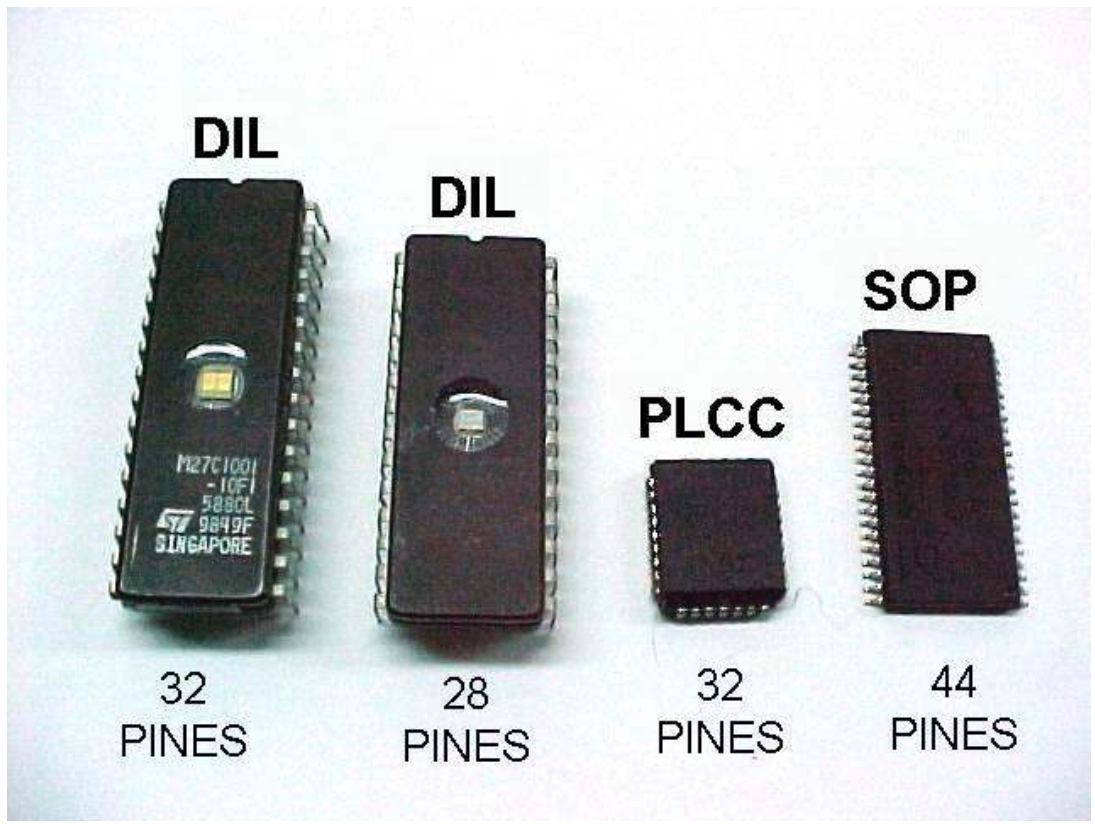
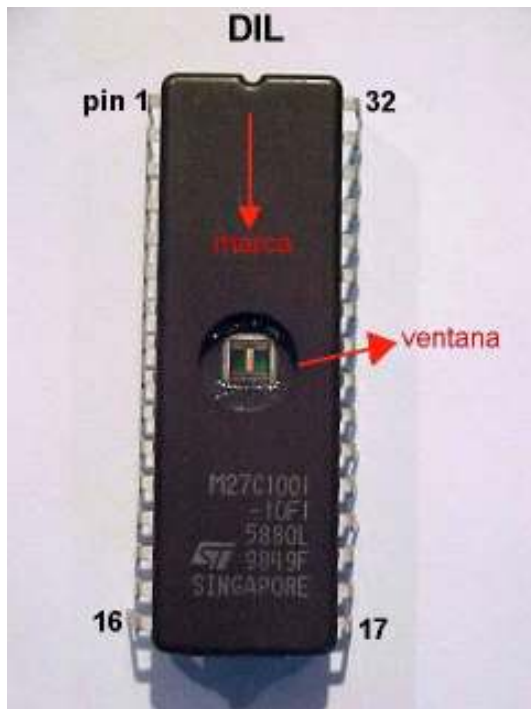


TIPOS DE MEMORIAS



Memoria tipo Dil.



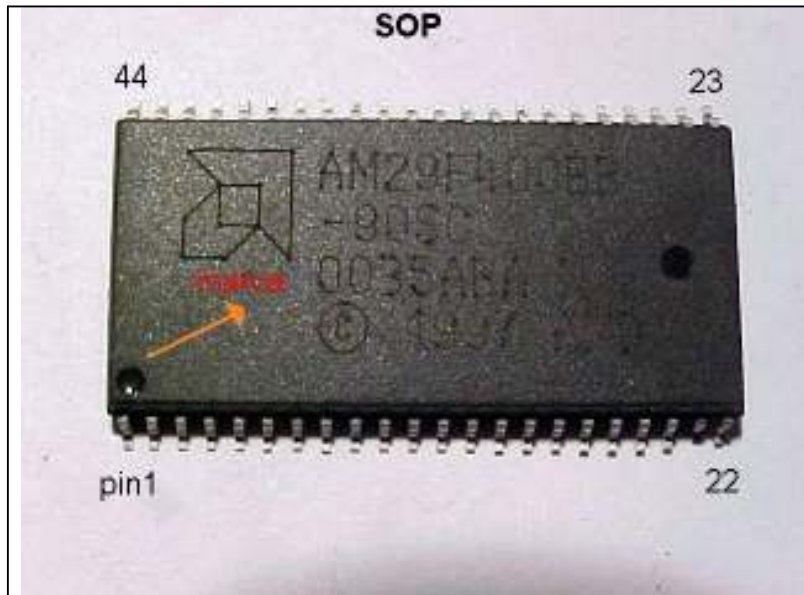
Este encapsulado fue uno de los **primeros** usados para los autos **equipados** con inyección electrónica **desde** el año 1990. Como se aprecia en la **figura** este posee 2 líneas con patas o **pin**es de acceso, estas pueden ser de 28 o **32** pin. Una marca o muesca se **puede** **observar** en su encapsulado, el mismo **muestra** la orientación de su propia **numeración**, y la ubicación del pin 1. **También** encontraremos una ventana en el medio del componente el cual nos **indica** que podemos borrar los datos de la misma con rayos ultravioletas. Este **proceso requiere** de un Borrador de **eprom** y el tiempo de ejecución es de **aproximadamente** 25 minutos.

Su montaje sobre la placa puede **realizarse** de 2 maneras distintas, **directamente** soldado **sobre** la placa o **puede** aparecer **también** montada sobre un **zócalo**, esto facilita su extracción y **posterior** trabajo.

Memorias tipo Plcc



Este encapsulado fue el segundo **utilizado** por las **terminales automotrices** el mismo cuenta con un tamaño reducido y la configuración de sus patas envuelve **los 4** lados a **diferencia de** su antecesor (**dil**) la cantidad de patas puede ser de 32, 44, 48 patas. Lo **particular** de esta **configuración** es que las patas o pin. 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 borrar eléctricamente para **luego** poder **reutilizarla**, este proceso se **realiza** con un programador de **memorias**. Sobre un lateral posee una **marca** que nos indica la posición pin 1. **Puede** ser encontrado montado **directamente** sobre la placa como **también** sobre un **zócalo**, esto facilita su **extracción** y manejo.

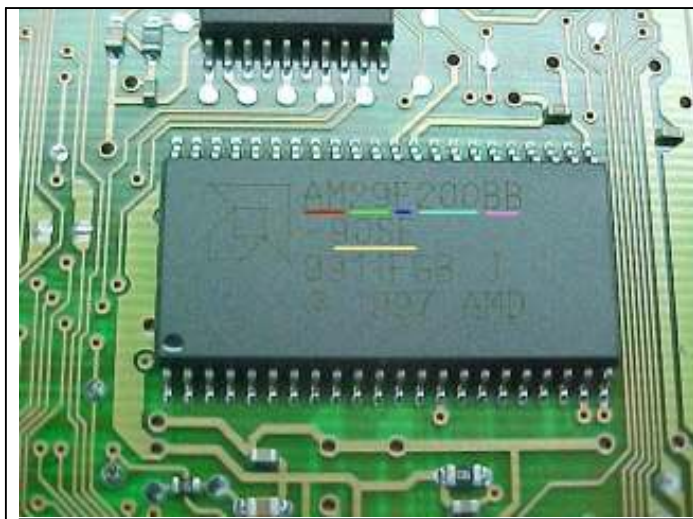


Memorias tipo Sop

Este encapsulado es uno de los mas **utilizados** en la actualidad, posee una **capacidad** de 2 hasta 32 megas en el **uso automotriz**, este encapsulado **logro** en su costo, capacidad y espacio **un** compromiso adoptado por los **fabricantes** de electrónica **automotriz**. La cantidad de pines utilizados es de **44** o 48 terminales, también posee una **marca** el cual nos indica la posición del pin 1.

CAPACIDADES DE LAS MEMORIAS

1. Interpretación de su nomenclatura



Rojo (AM) marca del fabricante.
Verde (29): familia a la cual **pertenece**.
Azul (F): flash memoria, memoria a la cual se borra **eléctricamente**.
Celeste (200): capacidad de la memoria, 2 megabytes.
Rosa (BB): tipo de **arquitectura interna**.
Amarillo (-90 sf): Tiempo o velocidad de acceso.

Verde (29): Para las memorias montadas en los **automóviles** se encuentran en su mayoría 2 tipos de familias la Nro. 27 y la Nro 29, cada familia **estructuralmente** cumple con ciertas condiciones como **velocidad** de acceso, condiciones de **circuitos internos** como **compuertas**, estructuras, **arquitectura**, etc. **Estas características** se aplican al uso al cual son sometidas.

Azul (F): Seguido del tipo de familia se **observa** la letra “C=5v” esto esta referido a su **alimentación** o letra “F=flash” esto esta referido a que su borrado puede realizarse **eléctricamente**. Esto se **realiza** mediante un **programador** de eproms, su ventaja Desarrollar las **tensiones** d la s memorias y beneficios **del** flash, borrado de los dil y flash.

Celeste (200): Los números **posteriores a su alimentación es la capacidad** que posee la memoria, lo **que** puede almacenar internamente y pueden ser **para las memorias Dil, Plcc, Sop**, aplicadas a los **automóviles:**

128k, 256k, 512k, **1024k, 2048k**, 4096k, 8192k

Equivalencias:

1024k	1Mega.
2048k	2Mega.
4096k	4Mega.
8192k	8Mega.

2. Tiempo de acceso

Amarillo (-90): El tiempo de acceso es el tiempo **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, si algunas localidades de la **memoria** se **alcanzan más rápidamente** que otras se **suele** tomar el valor promedio de todas ellas, se habla entonces del tiempo de acceso promedio.

Ej.: -10 ns = menor a 10 nanosegundos.

-120 ns = menor a 120 nanosegundos

1 nanosegundo = 1/1.000.000.000 (una milmillonésima) de Segundo.

3. logo del fabricante.



Usualmente cada proveedor de memorias **identifica** a su producto colocando un logo. Hay en el **mercado innumerables marcas** de fabricantes de **memorias:**

ST, AMD, ATMEL, INTEL, **FUJITSU**, NEC,etc

Cabe destacar que debido a los distintos **fabricantes de componentes**, existen **diferentes formas de colocar la información. Pero** en su mayoría **uno** puede ver a que **tipo de familia** pertenece, su **capacidad** y su velocidad de acceso. Algunos datos **perteneces** a datos **internos** de cada fabricante, **numero** de fabricación, lote, etc.

1001=1024=1 megabyte 001=1024=1megabyte

También existen manuales referidos a los distintos **tipos** de memorias y su aplicación, donde dan todas **las características** correspondientes de cada una de **ellas.**

[Estructura de funcionalidad de una memoria flash](#)

1. Ejercicios

EXTRACCION DE LAS MEMORIAS

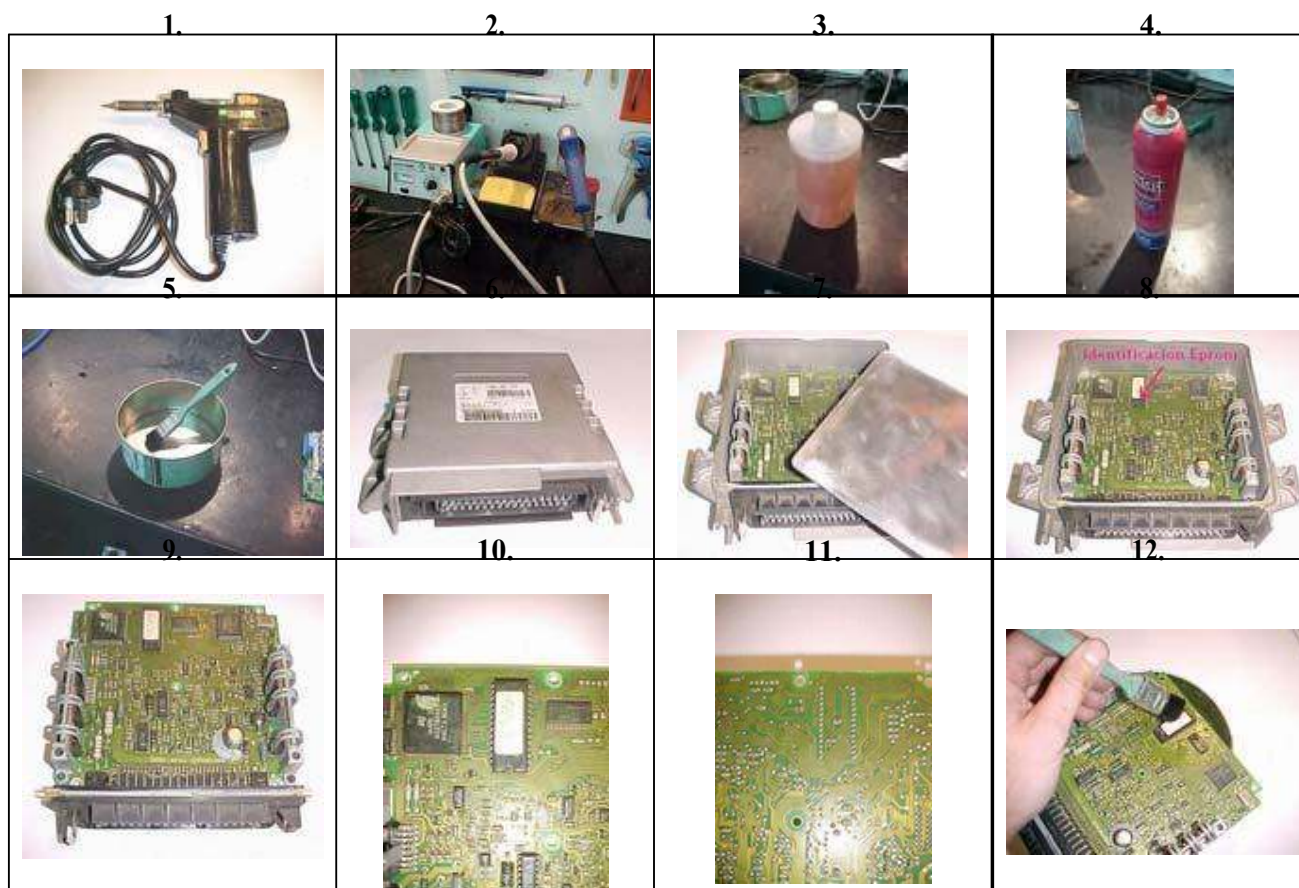
Haremos mención de los 3 casos **particulares** mas frecuentes en las Ecu **motor**.

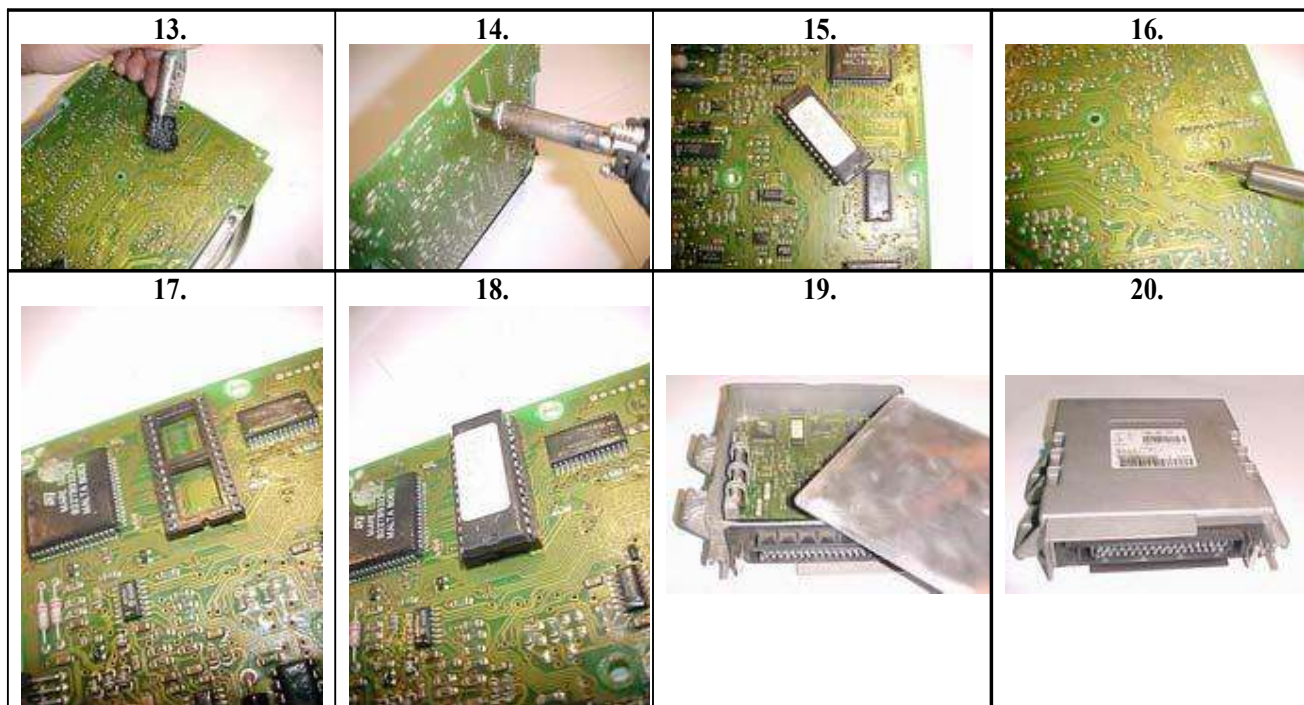
MEMORIAS DIL

Cabe destacar que es necesario **realizar** una buena **extracción** y conservar en buen estado le memoria y la placa madre.

Herramientas necesarias para poder **realizar** la **extracción de** la memoria DIL.

1. Desoldador con regulador de temperatura y bomba de vació.
2. Soldador con regulador de temperatura.
3. Liquido Flux.
4. Limpiador de impurezas
5. Recipiente con pincel y thinner de limpieza.
6. Ecu motor
7. Desmontar la placa pegada en la parte trasera de la ecu motor.
8. Ubicación de la **memoria** en la placa madre
9. Desmontar **totalmente** la placa madre de la **carcaza metálica**.
10. Observar que la memoria esta **soldada** sobre la placa madre, este trípode soldadura se realiza con **maquinas** robotizadas desde su fabricación.





11. Al girar la placa se observa **que los pines de la memoria DIL cruzan toda la placa madre**, esto **significa** que la **soldadura** esta **hecha** sobre las dos **fases** de la placa madre. Esto imposibilita **realizar la extracción con** un soldador común.

12. Con el recipiente **que** contiene de thinner de **limpieza** y el pincel **desmontar la cobertura** gomosa de protección que se observa en la placa madre.

13. Realizar el mismo procedimiento en las 2 fases **de la** placa madre. Luego rociar sobre los 2 lados **de** la placa con **FLUX liquido**.

14. Con el Desoldador con regulador de temperatura y bomba de vacío se procede a la extracción del **estaño** de las 2 fases de la placa esto se realiza **gracias** a que la punta del soldador es la que posee la **temperatura** necesaria para derretir el estaño (340 *C) de ambos **lados de** la placa y al presionar el **interruptor** del **soldador comienza a** succionar el estaño ya en estado liquido y se deposita en **un compartimiento** de reserva del desoldador. Este **procedimiento** se realiza según la cantidad de pines **de** la memoria.

15. Una vez desmontada de la placa podemos colocar un **zócalo dil** para proceder a extraer de la placa **la memoria** con más **facilidad** sin **producir ningún** daño sobre la **placa madre**.

16 / 17. Para soldar la memoria Dil **nuevamente** o el **zócalo** se realiza la soldadura del lado opuesto al **lado** componente, y con el Soldador con regulador de temperatura.

El estaño fluirá hacia **ambos** lados de la placa **realizando** el contacto necesario **para** la conducción en **ambos** lados de la placa.

18. Colocar la **memoria sobre el zócalo DIL**. Tener **precaución** de que coincidan en línea **todos** los **pines** de la memoria **en el momento de presionar** sobre el **zócalo**.

19 / 20. Proceder en orden inverso al desarmado.

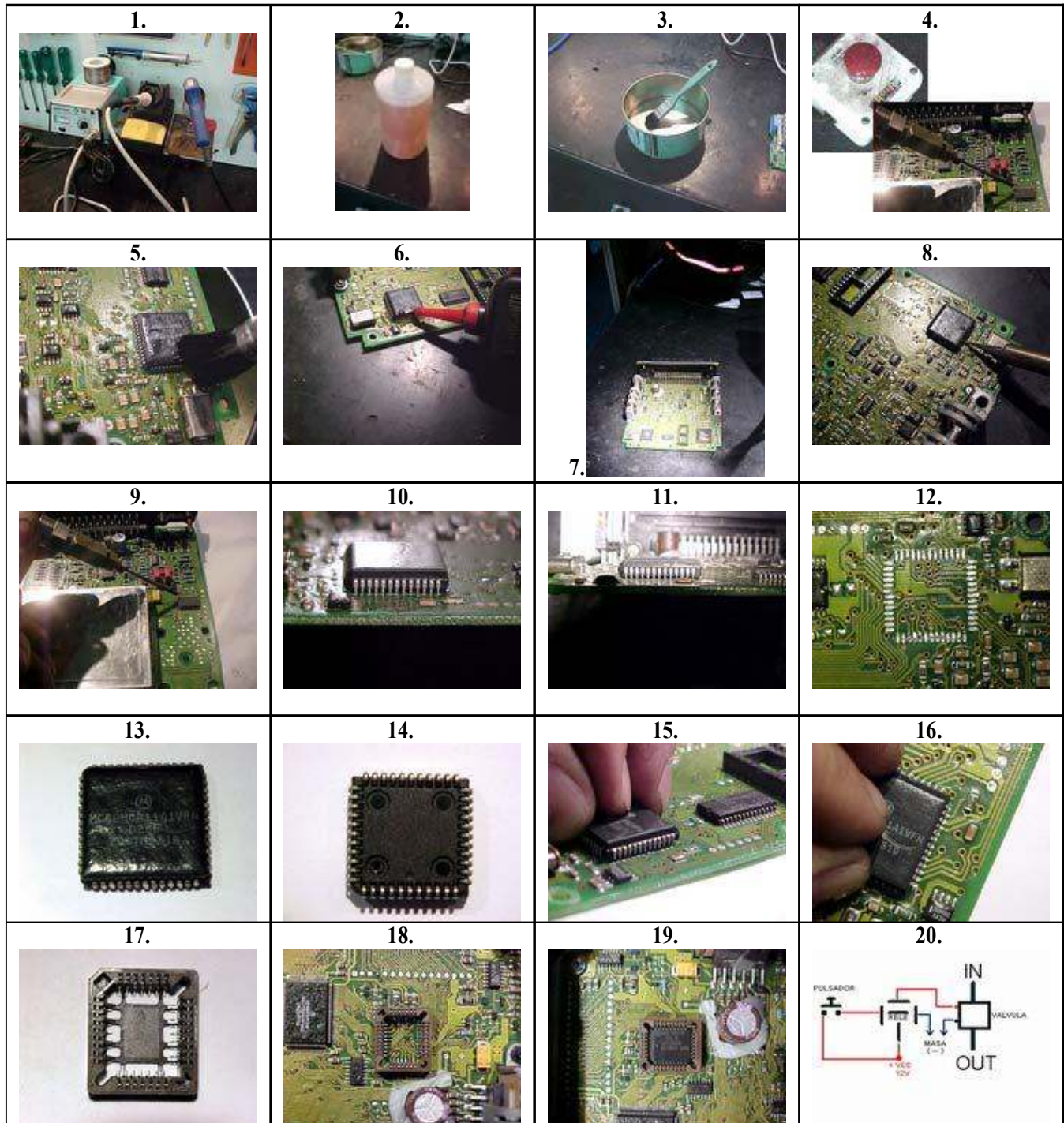
***El líquido flux se utilizara en caso de querer soldar la memoria eprom sin zócalo de extracción.**

MEMORIAS PLCC

A diferencia de las memorias **DIL**, la **extracción** puede **realizarse** con diferentes métodos y **Desoldadotes**, en el ejemplo a realizar lo haremos con un método económico y relativamente practico.

Herramientas necesarias para poder **realizar** la **extracción** de la memoria PLCC:

1. Soldador con regulador de temperatura.
2. Liquido Flux.
3. Recipiente con pincel y thinner de limpieza.



4. Pistola **de aire comprimido**, Pulsador y válvula.
5. Con el recipiente que contiene de thinner de **limpieza** y el pincel desmontar la cobertura **gomosa de protección** que se observa en la placa madre.
6. Rociar uno de los laterales con flux, con ayuda de **una** lampa de alta luminiscencia (dicroica foto 7).
- 8 y 9. Con el soldador calentar uno de los **terminales de** la memoria y al **producirse** el **calentamiento** del estaño presionar el pulsador de pie que permite **la liberación** del estaño residual dejando la unión **entre placa** y pin de la memoria libre para **su extracción**, realizar este procedimiento acorde a la **cantidad** de pines tenga la misma.
- 10 y 11. Se puede apreciar a medida que se libera **el estaño** de los pines como **queda libre para** su **posterior** extracción, este proceso se **realiza sobre** los cuatro lados de la **memoria**.
12. Una **vez** retirada la memoria , **quedara** al descubierto todas las **pistas** de conexión con la placa, **proceder** a limpiar **bien esta zona** con el thinner y **el pincel**, ya que cuenta con desperdicios de estaño y **flux**, de **esta manera** dejarla **preparada** para su posterior trabajo.
13. Con la memoria ya **desmontada proceder** también **a su limpieza** en especial **de todos** sus pines **liberándolos** de todo desperdicio de material.
14. Vista **inferior de** la memoria donde se aprecian **los pines** que **realizan** el contacto con la placa madre.
- 15 y 16. En el caso que se necesite la **sustitución del componente** o la reprogramación de este, se **procede con** el **comienzo** de su montaje, la tarea **principal** es la del **centrado de** la misma a la placa madre, **cuanto mas pines tenemos** mejor tendremos **que realizar** el procedimiento, en el momento que se logro el perfecto centrado, procedemos a soldar **solo 2** terminales **cruzados**, esto nos permitirá que el componente quede **perfectamente** centrado **sin** la **posibilidad** de su movimiento, seguido soldamos **todos** los restantes pines de la memoria.
17. E el caso de tener que desmontar el componente reiteradas veces es necesarios la implementación **de un zócalo plcc**, este nos permitirá poder **desmontar** y montar el **componente de** la placa sin dañar la placa ni la memoria.
18. El procediemito de soldado del zócalo plcc necesita muy poco aporte de material estaño, se puede **ver** el la **ilustración que** casi no se aprecia el **estaño**. Debemos tener en cuenta que tanto el **zócalo** como la memoria poseen una posición frente a la placa **madre**, teniendo que **realizar** el montaje de igual **manera** a su original **posición**.
19. Una **vez realizado** la incorporación del **zócalo** y la programación de la memoria con una leve **presión** insertar la memoria en su **zócalo** corroborando la estanqueidad de esta.
20. esquema eléctrico de cómo **realizar** el pulsador de **aire** a presión para **realizar** la extracción de las memorias plcc.

MEMORIAS SOP

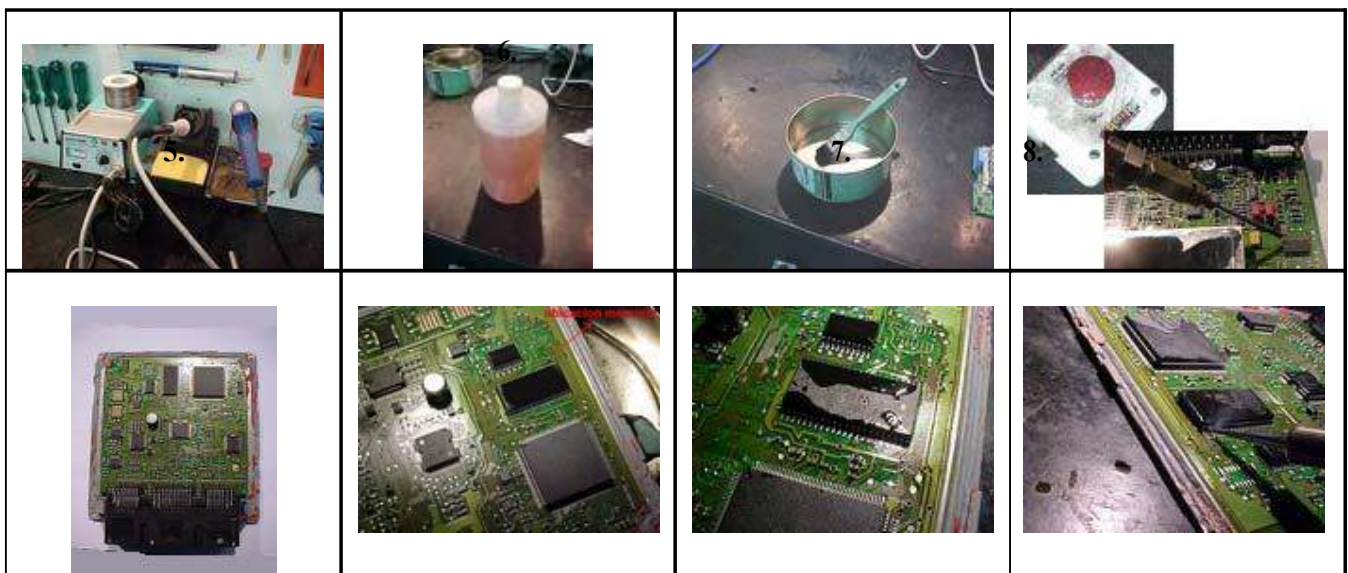
1. Soldador con regulador de temperatura.
2. Liquido Flux.
3. Recipiente con pincel y thinner de limpieza.
4. Pistola de aire comprimido, Pulsador y válvula.
5. Desprender la contratapa con precaución que se encuentra pegada con **sellador**.
- 6 y 7. Ubicar la memoria a desoldar y aplicar flux sobre **sus** terminales.
8. Con el soldador calentar uno de los terminales **de la** memoria y al **producirse** el calentamiento del **estaño** presionar el pulsador de pie que permite **la liberación** del estaño residual dejando la unión **entre** placa y pin de la memoria libre para su **extracción**, realizar este procedimiento acorde a la **cantidad** de pines tenga la misma.
9. Con una pequeña punta realizar una leve presión **donde** se encuentra la parte ya desprovista de **estaño** y verificar su despegue de la placa madre. **Realizar** este procedimiento en la otra cara de la **memoria**.
- 10 y 11. Una vez retirada la memoria , quedara al **descubierto** todas las pistas de conexión con la **placa**, proceder a limpiar bien esta zona con el **thinner** y el pincel, ya que cuenta con desperdicios de **estaño** y flux, de esta manera dejarla preparada para su posterior trabajo.
- 12 y 13. En el caso que se necesite la **sustitución del componente** o la reprogramación de este, se **procede con** el **comienzo** de su montaje, la tarea **principal** es la del **centrado de** la misma a la placa **madre**, **cuanto mas pines tenemos** mejor tendremos **que realizar** el procedimiento, en el momento que **se logro** el perfecto centrado, procedemos a soldar **solo 2** terminales **cruzados**, esto nos permitirá que **el componente** quede **perfectamente** centrado **sin** la **posibilidad** de su movimiento, seguido soldamos **todos** los restantes pines de la memoria.

1.

2.

3.

4.





14. En el supuesto caso que no **necesitemos** la **información del** componente podemos optar **por realizar** este trabajo, con una **pinza** de corte de pequeña **medida**, cortaremos una por **una** las patas o pines del **componente** , de esta manera no tendremos que **desoldar** el componente y **los** tiempos de trabajo son **mas cortos**.

15. Una **vez desmontado el componente** sobre la placa madre quedara un resto de los pines de la **memoria**.

16. Con el soldador barrer con el resto de pines **que quedaron** sobre la placa, y dejar limpio para **proceder** al siguiente **paso**.

17. En caso de querer montar un zócalo sop, es **necesario colocar** entre la placa y el **zócalo** unos pines **de ensamble**, esto permite **insertar el zócalo** sop y **realizar** la conducción entre la memoria y la placa **madre**.

18. Vista **superior** del **zócalo**.

19. Montar la memoria sobre el **zócalo**, centrarla y luego soldarla.

20. Con la memoria ya soldada al zócalo y los **pinados** de adaptación soldados a la placa madre, **insertar** el zócalo.

2. Practica extracción de **memorias**.

PROGRAMADORES DE EPROMS

Bien, ahora podemos extraer **cualquier** memoria de la placa madre del Ecu Motor, ahora tendremos que **realizar el procedimiento** de leer la **información contenida en la memoria**. Esto se **realiza mediante un programador** de eprom o memorias.

Existen **distintos** tipos y calidades de **programadores** de eproms en el mercado, estos pueden medirse por su **velocidad**, capacidad de manejo de distintos **componentes**, **espacio físico**, la **posibilidad de su manejo** portátil, su úpate automático, etc.



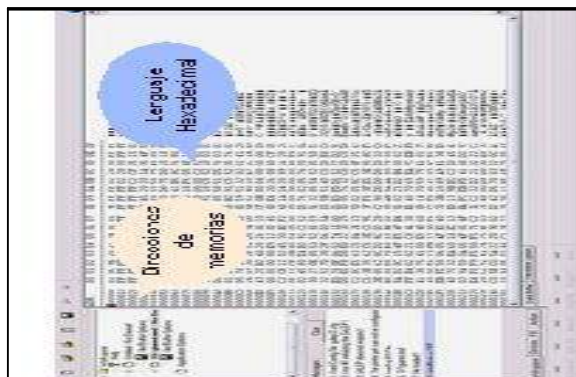
Programadores de banco

Son aquellos que su **utilización** es para **medianas** y grandes **producciones** de trabajo, su gran velocidad permite rápidamente **programar** y borrar memoria en **segundos**, su conexionado es **mediante puerto** paralelo o serial, **permite identificación automática** el **componente** a leer, su tamaño es **considerable**, y su alimentación es a **una red eléctrica**(110v, 220v).



Programadores portátiles

Su tamaño es reducido, se utiliza para trabajos de baja producción, la **velocidad** esta limitada, puede programar y borrar memorias en **minutos**, permite su traslado y la **posibilidad** de mediante pilas poder **realizar** trabajos en cualquier **lugar**, no posee lectura automática del **componente**, su conexionado es por puerto paralelo o **usb**.



Cada programador esta acompañado de su propio **soft** de trabajo, todos los soft modernos trabajan en **entorno** Windows y su **interpretación** es muy similar **uno** al otro.

En su entorno **principal** se **encuentra** la **matriz** de **datos hexademal**, las **direcciones** correspondientes a **la memoria**, su work espace o lugar de trabajo en **al PC**, y la información de trabajo del programador.

Ahora que ya **realizamos** el **desmontaje** de la **memoria** de la ecu, **procederemos a** insertar la memoria **en** el programador, y poder leer la información **dentro** contenida. Podemos **insertar** las memorias **dil** **directamente** sobre el **programador** pero para las **memorias plcc o sop** tendremos que **utilizar** unos **adaptadores** para su lectura.

3. Ejercicio trabajo con el programador, lectura memorias dil, plcc, sop, interpretación su almacenamiento.

BORRADORES DE EPROMS



Este dispositivo se encarga de mediante rayos ultravioletas y una **permanencia programada** borrar las **memorial DIL**, este tipo de **memorias poseen como** vimos anteriormente una ventana de acceso la **cual** al recibir borra **los** datos internos de la **memoria**, su exposición esta calculada entre 20 y 30 **minutos** para su completo vaciado.

De estos programadores **existen** los portátiles que **funcionan** a pila y los **de banco** que se **conectan a** la línea eléctrica, **también existen** distintos tipos de **tamaño**, con la capacidad de poder borrar varias **memorias** juntas.

ZOCALOS ADAPTADORES PARA PROGRAMACION

En todos los programadores de memorias se **encuentra** un único formato para insertar el componente y este es el DIL, para poder insertar los **distintos componentes** en el programador es necesario la **utilización** de diversos adaptadores para **realizar** una **acople** entre la cantidad de **terminales** de cada **componente** y la adaptación de su electrónica **propia**. Para esto cada programador desarrollo los **adaptadores correspondientes a cada caso**. Existen **también** adaptadores del tipo universal.



Lectura Memorias DIL



Para la inserción de la memorias **dil** en el programador no es necesario **ningún** adaptador, solo hay que **respetar** la **posición marcada** en el **programador** con la posición de **la memoria**, esta hace coincidir la **numeración** de los pines. Existe una pequeña palanca que nos facilita que **la memoria permanezca** bien conectada al **programador**.



Vista Zócalo Plcc-Dil

Lectura Memorias PLCC

Para la inserción de las memorias **plcc** es **necesario la utilización** de un adaptador entre el **programador** y la memoria. Para los diferentes **tamaños de memorias plcc** hará **falta un adaptador** diferente. Sobre el **zócalo** adaptador **estará** la marca que identifica la colocación **correcta** para su lectura.

	 <p data-bbox="532 432 786 464">Vista Zócalo sop-dil</p>	<p data-bbox="1003 96 1305 128" style="text-align: center;"><u>Lectura Memorias SOP</u></p> <p data-bbox="841 163 1430 499"> Para la inserción de las memorias sop también será necesario la utilización de un adaptador entre el programador y la memoria, el tamaño de este adaptador es uno de los mas grandes debido a el trabajo que tiene que realizar para efectuar la lectura correspondiente, este tipo de adaptador permite leer una gran variedad de memorias sop. En la hoja correspondiente al adaptador indica su posición en el programador y la correcta colocación de sus jumpers. </p>
--	--	---

INTERPRETACION DE LOS LENGUAJES Y SUS EQUIVALENTES

La forma de interpretación de su lectura la vamos a realizar se **podría realizar** en **distintos sistemas numéricos**, en general la información contenida en la memoria la **traduciremos** en lenguaje **hexadecimal**, a **continuación** veremos una breve **descripción** de los sistemas numéricos y su equivalencia.

SISTEMAS NUMERICOS

El **sistema numérico que utilizamos** a diario es el **sistema decimal**, pero este **sistema** no es **conveniente para** las máquinas debido a que la información se maneja codificada en forma de bits prendidos o **apagados**; esta forma de codificación nos **lleva a la necesidad** de conocer el **cálculo** posicional que nos **permita expresar** un número en **cualquier** base que lo necesitemos.

Es posible representar un número determinado en **cualquier** base mediante la siguiente formula:

Donde n es la posición del **dígito empezando de derecha** a izquierda y numerando a partir de cero. D es el **dígito** sobre el **cual operamos** y B es la base numérica empleada.

Convertir números binarios a decimales

Trabajando en el lenguaje ensamblador nos **encontramos** con la **necesidad** de convertir números del **sistema binario**, que es el **empleado por** las **computadoras**, al sistema decimal utilizado por las **personas**.

El sistema binario está basado **en** únicamente dos condiciones o estados, ya sea encendido (1) o **apagado** (0), por lo tanto su base es dos.

Para la conversión podemos **utilizar** la formula de valor posicional:

Por ejemplo, si tenemos el número binario 10011, tomamos de derecha a izquierda cada dígito y lo multiplicamos por la base elevada a la nueva posición que ocupan:

Binario: 1 1 0 0 1

Decimal: $1*2^0 + 1*2^1 + 0*2^2 + 0*2^3 + 1*2^4$

$= 1 + 2 + 0 + 0 + 16 = 19$ decimal.

El carácter ^ es **utilizado** en computación como símbolo de potenciación y el carácter * se usa para representar la multiplicación.

Convertir números decimales a binarios

Existen varios métodos de conversión de números **decimales** a binarios; aquí' solo se analizará uno.

Naturalmente es **mucho** más fácil una conversión con **una calculadora científica**, pero no siempre se **cuenta** con ella, **así** que es conveniente **conocer por lo menos** una forma manual para **hacerlo**.

El método que se explicará **utiliza** la división sucesiva entre dos, **guardando** el residuo como dígito **binario** y el resultado como la **siguiente cantidad** a dividir.

Tomemos como ejemplo el **número 43 decimal**.

$43/2 = 21$ y su residuo es 1

$21/2 = 10$ y su residuo es 1

$10/2 = 5$ y su residuo es 0

$5/2 = 2$ y su residuo es 1

$2/2 = 1$ y su residuo es 0

$1/2 = 0$ y su residuo es 1

Armando el número de abajo hacia arriba **tenemos** que el resultado en binario es 101011

Sistema hexadecimal

En la base hexadecimal tenemos 16 dígitos que van **de 0** a 9 y de la letra A hasta la F (estas letras **representan** los números del 10 al 15). Por lo tanto, contamos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F.

La conversión entre numeración binaria y hexadecimal es sencilla. Lo primero que se hace para una **conversión** de un número binario a hexadecimal es **dividirlo** en grupos de 4 bits, **empezando de derecha** a izquierda. En caso de que el último grupo (**el que** quede más a la **izquierda**) sea menor de 4 **bits** se rellenan los faltantes con ceros.

Tomando como ejemplo el número binario 101011 **lo dividimos** en grupos de 4 bits y nos queda:

10; 1011

Rellenando con ceros el último **grupo** (el de la izquierda):

0010; 1011

Después tomamos cada grupo como un número **independiente** y **consideramos** su valor en decimal:

0010 = 2; 1011 = 11

Pero como no podemos **representar** este número hexadecimal como **211** porque sería un **error**, **tenemos que** sustituir todos los valores mayores a **9** por su respectiva representación en hexadecimal, **con** lo que obtenemos:

2BH (Donde la H representa la base hexadecimal)

Para convertir un número de **hexadecimal** a binario solo **es necesario invertir estos** pasos: se toma el **primer dígito** hexadecimal y se **convierte a binario**, y luego el **segundo**, y así **sucesivamente** hasta **completar** el número.

INTERPRETACION Y PROCESO DE TRABAJO

Bien, ahora que ya pudimos leer **la información** de la memoria, procederemos **a** guardar la misma en **un** directorio por nosotros **asignados**. Las extensiones en las que se guardan estos archivos **son siempre** en Binario, por **consecuencia** su **extensión** será por ejemplo: nombre del **archivo.bin**

Por consiguiente tendremos que encontrar la manera **de** conseguir la **interpretación** de la **información** **contenida** dentro de la memoria, **sabemos** que **la misma hasta** el momento podemos apreciarla en **idioma hexadecimal mediante** la **lectura** del **programador**. Se nos haría muy trabajoso interpretar y **poder llegar** a encontrar los **datos contenidos** en **la memoria** en forma hexadecimal observándolos en la **matriz de** un **software** de un programador, su contenido es **demasiado** largo de recorrer y **no podríamos** obtener de alguna manera una forma de **encontrar** los **datos** que a nosotros nos interesan y **proceder** a su **modificación**.

Los datos contenidos en **la memoria** son los **utilizados** por el **microprocesador** para **informar un instante** de conducción, esto sería, encontrar **para un** determinado giro de motor, vacío de motor, **temperatura de** motor, **temperatura del** aire **de admisión**, posición del acelerador motor, **estado** del **oxígeno** de los gases de escape y **alguna** otra **información que** la ECU considere importante, poder **determinar** y encontrar un grado de avance para **el encendido** y una dosificación de combustible acorde para ese **instante**. Este proceso se **realiza** por cada instante a una velocidad por nosotros **inalcanzable** de interpretar, este trabajo es la **búsqueda** de conformar y encontrar la mejor función **del** automóvil, cada fabricante **vuelca** sobre la memoria los datos correspondientes para que pueda **realizar** y encontrar dentro de **esta toda la información las distintas posibilidades** que puede **producirse** en el manejo del automóvil, **esto significa** que para cada conjunto de datos **recibidos** de los **distintos** sensores de motor analizara, buscara en **la memoria** y procesara **la mejor** alternativa para ese instante, este trabajo se realiza **indefinidamente** en el tiempo mientras **el** auto permanezca en **marcha**.

Por el momento tenemos como **datos validos las direcciones** dentro de las memorias, **están** serán de **mayor** o menor cantidad dependiendo de su capacidad:

Capacidad de la memoria (K)	Tamaño (Bytes)	Posiciones Hexadecimales
256	32768	0000-7FFF
512	65536	0000-FFFF
1024	131072	0000-1FFFF
2048	262144	0000-3FFFF
4096	524288	0000-7FFFF
8192	1048576	0000-FFFFFF

