



## El fichero HEX explicado

*Veremos cómo está compuesto el fichero .hex que se genera al compilar nuestro programa. Esta información será muy útil sobre todo para los nuevos diseñadores de grabadores de PIC's o para los que estén entrando en el tema de los Bootloaders. Ambos deben empezar por conocer esto de los .hex*

*Por: Diego "RedPic" Márquez - <http://picmania.garcia-cuervo.net>*

### Introducción

Cada vez que compiláis u ensambláis un programa fuente vuestro compilador u ensamblador genera un fichero .hex cuyo contenido se corresponde exactamente con lo que ha de ser grabado en la memoria de programa (o EEPROM) del microcontrolador.

Fijaos que he dicho "se corresponde" y no que sea exactamente lo que ha de ser grabado en el PIC, no es una "imagen" de la memoria de programa del PIC, sino una serie de instrucciones que el grabador de PIC's o el Bootloader que utilicemos sabe interpretar y por lo tanto grabar lo que corresponde exactamente en su sitio. Más adelante veremos cómo están codificadas las direcciones y su contenido.

### Un poco de Historia

El formato .hex es del fabricante de micros INTEL que lo inventó allá por los años 70 del siglo pasado para usarlo exactamente para lo mismo que nosotros lo estamos usando ahora pero para sus microprocesadores 8085 y otros cacharros antediluvianos por el estilo y desde entonces está en uso. Muchos otros fabricantes lo han adoptado como propio y otros lo han copiado, cambiando esto y aquello, para al final hacer lo mismo (por ejemplo el SRecord de Motorola y otros).

Aunque hay tres tipos de ficheros HEX: de 8 bits, de 16 bits y de 32 bits también llamados I8HEX, I16HEX e I32HEX respectivamente en esta nota vamos a ver solo el de 8 bits que es el que más utilizamos para nuestros PIC's 16F y 18F, el resto son muy parecidos pero no iguales.

*En el fondo un fichero .hex no es más que una lista de direcciones de memoria y lo que contiene cada una de estas posiciones.*

### Descripción

Un fichero .hex es un fichero de texto. Por lo tanto puede ser editado con un notepad o similar. Consiste en una serie de líneas consecutivas que empiezan siempre por el carácter ":" (dos puntos) salvo los comentarios que usan ";" (punto y coma) y terminadas en [0x0D][0x0A] (Fin de línea, Retorno de Carro)

Todos números (longitudes, direcciones y datos) están expresados en **HEXADECIMAL** mediante sus caracteres **ASCII** correspondientes.

Una muestra de su apariencia es:

```
:020000040000FA
:1000000043EF00F0EA6A070EE96EEF500DE0060ECE
:10001000016E006A002EFED7012EFBD77B0E006E0C
:1000200002EFED7EF2EF3D7000C0990099208524C
:1000300002E0098001D0098207C003F00650D8B45D
:100040000706060603101EE0000E0AB2010E0B6E34
:10005000000E09B0010E0B24016EE8B002D0819CA5
:1000600001D0818C000E0AB2010E0B6E000E09B297
:10007000010E0B24016EE8B002D0819C01D0818C6E
:100080000A2ADAD7000CF86AD09EEA6AE96AC150F7
:10009000C00B0F09C16E070EB46E040E066EFA0E89
:1000A000076EB0DF062EFBD7076A190E066E086AC8
:1000B000BCDF040E066EFA0E076EA4DF062EFBD719
:1000C000076A190E066E010E086EAFDFE6D7030051
:020000040030CA
:0E000000000C1E1F008381000FC00FE00F4098
:00000001FF
;PIC18F4550
```

La estructura de una línea es:

- **Start code** Un caracter, ":" para líneas con contenido, ";" para comentarios.
- **Byte count** Dos caracteres HEX que indican el número de datos en **Data**.
- **Address** Cuatro caracteres HEX para la dirección del primer dato en **Data**.
- **Record type** Dos caracteres HEX que indican el tipo de línea, de 00 a 05. (ver mas abajo)
- **Data** Secuencia de 2 \* n caracteres HEX correspondientes a los **Byte count** datos definidos antes.
- **Checksum** Dos caracteres HEX de Checksum calculado según el contenido anterior de la línea en la forma: El byte menos significativo del complemento a dos de la suma de los valores anteriores expresados como enteros los caracteres hexadecimales (sin incluir ni el **Start code** ni al él mismo)

Cada línea puede expresar según su **Record type**:

- **00 data record**: Línea de datos, contiene la dirección del primer dato y la secuencia de datos a partir de ésta.
- **01 End Of File record**: Línea de Fin del Fichero HEX. Indica que se han acabado las líneas de datos. Usualmente es ":0000001FF"
- **02 Extended Segment Address Record**: Usado para procesadores 80x86 (No nos interesa aquí)
- **03 Start Segment Address Record**: Usado para procesadores 80x86 (No nos interesa aquí)
- **04 Extended Linear Address Record**: Si las líneas de datos que sigan a ésta necesitan una dirección mayor que la de 16 bits ésta línea aporta los otros 16 bits para completar una dirección completa de 32 bits. Todas las líneas que sigan a esta deben completar su dirección con hasta los 32 bits con el contenido de la última línea de tipo **04**
- **05 Start Linear Address Record**: Usado para procesadores 80386 o superiores (No nos interesa aquí)



**Nuestro Ejemplo** Podemos así entonces interpretar nuestro ejemplo anterior de la siguiente forma:

```
: 02 0000 04 0000 FA
```

-> Línea relevante, con dos bytes de información, de tipo 04 : luego las direcciones siguientes se complementan a 32 bits con 0x0000, 0xFA es el checksum.

```
: 10 0000 00 43 EF 00 F0 EA 6A 07 0E E9 6E EF 50 0D E0 06 0E CE
```

-> Línea relevante, con 16 bytes de información, de tipo 00 : así que hay que escribir 0x43 (en 0x00000000), 0xEF (en 0x00000001), ... , 0x0E (en 0x0000000F). 0xCE es el checksum.

```
: 10 0010 00 01 6E 00 6A 00 2E FE D7 01 2E FB D7 7B 0E 00 6E 0C
```

-> Línea relevante, con 16 bytes de información, de tipo 00 : así que hay que escribir 0x01 (en 0x00000010), 0x6E (en 0x00000011), ... , 0x6E (en 0x0000001F). 0x0C es el checksum.

...

...

...

```
: 02 0000 04 0030 CA
```

-> Línea relevante, con dos bytes de información, de tipo 04 : luego las direcciones siguientes se complementan a 32 bits con 0x0030, 0xCA es el checksum.

```
: 0E 0000 00 00 0C 1E 1F 00 83 81 00 0F C0 0F E0 0F 40 98
```

-> Línea relevante, con 14 bytes de información, de tipo 00 : así que hay que escribir 0x00 (en 0x00300000), 0x0C (en 0x00300001), ... , 0x40 (en 0x0030000D). 0x98 es el checksum.

```
:00000001FF
```

-> Línea relevante. Fin de Fichero HEX

```
;PIC18F4550
```

-> Línea irrelevante. Comentario.

Eso es todo por este documento. Espero que os sea de ayuda.