# What is the Intel HEX file format?

The Intel HEX file is an ASCII text file with lines of text that follow the Intel HEX file format. Each line in an Intel HEX file contains one HEX record.

These records are made up of hexadecimal numbers that represent machine language code and/or constant data. Intel HEX files are often used to transfer the program and data that would be stored in a ROM or EPROM. Most EPROM programmers or emulators can use Intel HEX files.

## Record Format

An Intel HEX file is composed of any number of HEX records. Each record is made up of five fields that are arranged in the following format:

```
:llaaaatt[dd...]cc
```

Each group of letters corresponds to a different field, and each letter represents a single hexadecimal digit. Each field is composed of at least two hexadecimal digits-which make up a byte-as described below:

: is the colon that starts every Intel HEX record.

ll is the record-length field that represents the number of data bytes (dd) in the record.

aaaa is the address field that represents the starting address for subsequent data in the record.

tt is the field that represents the HEX record type, which may be one of the following:

00 - data record

01 - end-of-file record

02 - extended segment address record

04 - extended linear address record

dd is a data field that represents one byte of data. A record may have multiple data bytes. The number of data bytes in the record must match the number specified by the ll field.

cc is the checksum field that represents the checksum of the record. The checksum is calculated by summing the values of all hexadecimal digit pairs in the record modulo 256 and taking the two's complement.

## Data Records

The Intel HEX file is made up of any number of data records that are terminated with a carriage return and a linefeed. Data records appear as follows:

```
:10246200464C5549442050524F46494C4500464C33
```

where:
10 is the number of data bytes in the record.
2462 is the address where the data are to be located in memory.
00 is the record type 00 (a data record).
464C...464C is the data.
33 is the checksum of the record.

Extended Linear Address Records (HEX386)
Extended linear address records are also known as 32-bit address records and HEX386 records. These records contain the upper 16 bits (bits 16-31) of the data address. The extended linear address record always has two data bytes and appears as follows:
:02000004FFFFFC

where:
02 is the number of data bytes in the record.
0000 is the address field. For the extended linear address record, this field is always 0000.
04 is the record type 04 (an extended linear address record).
FFFF is the upper 16 bits of the address.
FC is the checksum of the record and is calculated as
01h + NOT(02h + 00h + 00h + 04h + FFh + FFh).
When an extended linear address record is read, the extended linear address stored in the data field is saved and is applied to subsequent records read from the Intel HEX file. The linear address remains effective until changed by another extended address record.

The absolute-memory address of a data record is obtained by adding the address field in the record to the shifted address data from the extended linear address record. The following example illustrates this process.

| | |
|---|---|
| Address from the data record's address field | 2462 |
| Extended linear address record data field | FFFF |
| | -------- |
| Absolute-memory address | FFFF2462 |

Extended Segment Address Records (HEX86)
Extended segment address records-also known as HEX86 records-contain bits 4-19 of the data address segment. The extended segment address record always has two data bytes and appears as follows:
:020000021200EA

where:
02 is the number of data bytes in the record.

0000 is the address field. For the extended segment address record, this field is always 0000.
02 is the record type 02 (an extended segment address record).
1200 is the segment of the address.
EA is the checksum of the record and is calculated as
01h + NOT(02h + 00h + 00h + 02h + 12h + 00h).

When an extended segment address record is read, the extended segment address stored in the data field is saved and is applied to subsequent records read from the Intel HEX file. The segment address remains effective until changed by another extended address record.

The absolute-memory address of a data record is obtained by adding the address field in the record to the shifted-address data from the extended segment address record. The following example illustrates this process.

| | |
|---|---|
| Address from the data record's address field | 2462 |
| Extended segment address record data field | 1200 |
| | -------- |
| Absolute memory address | 00014462 |

End-of-File (EOF) Records
An Intel HEX file must end with an end-of-file (EOF) record. This record must have the value 01 in the record type field. An EOF record always appears as follows:
:00000001FF

where:
00 is the number of data bytes in the record.
0000 is the address where the data are to be located in memory. The address in end-of-file records is meaningless and is ignored. An address of 0000h is typical.
01 is the record type 01 (an end-of-file record).
FF is the checksum of the record and is calculated as
01h + NOT(00h + 00h + 00h + 01h).

Example Intel HEX File
Following is an example of a complete Intel HEX file:

```
:10001300AC12AD13AE10AF1112002F8E0E8F0F2244
:10000300E50B250DF509E50A350CF5081200132259
:03000000020023D8
:0C002300787FE4F6D8FD7581130200031D
:10002F00EFF88DF0A4FFEDC5F0CEA42EFEEC88F016
:04003F00A42EFE22CB
```

:00000001FF

Hex文件的INTEL格式:这是Intel公司提出的按地址排列的数据信息,数据宽度为字节,所有数据使用16进制数字表示.

这是一个例子:
:10008000AF5F67F0602703E0322CFA92007780C361
:1000900089001C6B7EA7CA9200FE10D2AA00477D81
:0B00A00080FA92006F3600C3A00076CB
:00000001FF

第一行,":"符号表明记录的开始. 后面的两个字符表明记录的长度,这里是10h. 后面的四个字 符给出调入的地址,这里是0080h. 后面的两个字符表明记录的类型;
0 数据记录 1 记录文件结束 2 扩展段地址记录 3 开始段地址记录 4 扩展线性地址记录 5 开始线性地址记录
后面则是真正的数据记录, 最后两位是校验和检查,它加上前面所有的数据和为0.

最后一行特殊,总是写成这个样子.

扩展Intel Hex的格式(最大1M): 由于普通的Intel的Hex记录文件只能记录64K的地址范围,所以大于64K的地址数据要靠扩展Intel Hex格式的文件来记录.对于扩展形式Hex文件,在每 一个64K段的开始加上扩展的段地址规定,下面的数据地址均在这个段内,除非出现新的段地址定义.

一个段地址 定义的格式如下:
起始符 长度 起始地址 扩展段标示 扩展段序号 无用 累加和
: 02 0000 02 3000 EC
段地址的标识符是第四组数据02,表示扩展地址段的定义,再后面的以为HEX数表示段的数目,上面的定义为3,表示段地址是3,所以下面的数据地址是3 + XX(XX是64K段内的地址)

==========

Intel HEX 文件是用来保存单片机或其他处理器的目标程序代码的文件,它保存物理程序存储器中的目标代码的映像,以便编程器和仿真器调用.绝大多数编程器都支持 Intel HEX 格式。

下面是一个 Intel HEX 文件用记事本打开后看到的内容:

```
:020000040000FA
:1000000018F09FE518F09FE518F09FE518F09FE5C0
:1000100018F09FE5805F20B9F0FF1FE518F09FE51D
:10002000C000000040000000440000004800000044
:100030004C00000000000000000000005000000024
......
:103020005C300000A8E60040000000005C300000BA
:10303000000001400000000048300000000000000D7
:1030400014000040 94E6000032FFF0FFE8030000A7
:0C30500064000000FFFFFFFF0100000013
:00000001FF
```

Intel HEX 文件是文本行的 ASCII 文本文件,文件内容全部由可打印的 ASCII 字符组成,可以用记事本打开.

Intel HEX 由一条或多条记录组成,每行一个记录,每条记录都以冒号":"开始,以回车(0DH)和换行(0AH)结束.

除":"外,每条记录有五个域,每一域由 2N(N>=1)个 HEX 字符组成,格式如下

```
:[AA][BBBB][CC][DD....DD][EE]
```

其中:[LL]　　:表示该记录的实际数据的长度;

[ZZZZ]　:表示该记录所包含的数据在实际的存储区中的起始地址;

[TT]　　:为该记录的类型;

[SS....SS]:为该记录的实际数据,由 2N(N>=1)个 HEX 字符组成,该域的长度应当与[LL]域所指出长度一致.

[RR]　　:为该记录的数据校验和.

例如对上面例子中的第一行:

:020000040000FA

用"["和"]"分开后如下:　　:[02][0000][04][0000][FA]

[02]　:该记录的实际数据的长度[LL]为 2 个字节(4 个 HEX 字符);

[0000]:该记录所包含的数据在实际的存储区中的起始地址[ZZZZ]为 0000H;

[04]　:该记录的类型[TT]为 04——扩展线性地址;

[0000]:该记录的实际数据[SS....SS];

[FA]　:该记录的数据校验和[RR];

对上面例子中的倒数第三行

:10303000000001400000000048300000000000000D7

用"["和"]"分开后如下:

:[10][3030][00][00000140000000004830000000000000][D7]

[10] :该记录的实际数据的长度[LL]为16D(10H)个字节(20H个HEX字符);

[3030]:该记录所包含的数据在实际的存储区中的起始地址[ZZZZ]为3030H;

[00] :该记录的类型[TT]为 00——数据(实际要烧写到存储器中的数据);

[0000]:该记录的实际数据[SS....SS];

[FA] :该记录的数据校验和[RR];

常见的记录类型如下:

00 :数据记录.表示该记录所包含的数据为实际要烧写到存储器中的数据。

01 :文件结束记录.表示该记录为本文件的最后一个记录。

02 :扩展段地址记录.表示该记录所包含的数据为段地址。

04 :扩展线性地址记录。表示该记录所包含的数据为线性地址。

校验和的计算规则:

以字节(2个HEX字符)为单位，除“:”以外，当前行所有数据的和为00H.注意对和只取低8位.

例如对上面例子中的第一行:

:020000040000FA

02 00 00 04 00 00 FA

02H+00H+00H+00H+04H+00H+00H+00H+FAH=100H

对上面例子中的倒数第三行

:10303000000001400000000048300000000000D7

10 30 30 00 00 00 01 40 00 00 00 00 48 30 00 00 00 00 00 00 D7

10H+30H+00H+00H+00H+30H+01H+40H+00H+00H+00H+00H+48H+30H+00H+00H+00H+00H+00H+00H+D7H=200H

扩展线性地址:

当一个扩展线性地址记录被读到后,扩展线性地址将被保存并应用到后面从Intel HEX文件中读出的记录,这个扩展线性一直有效,直到读到下一个扩展线性地址记录.

绝对地址与扩展线性地址的关系如下：

绝对地址=数据记录中的地址[ZZZZ]+移位后的扩展线性地址

扩展段地址记录

当一个扩展段地址记录被读到后,扩展段地址将被保存并应用到后面从Intel HEX文件中读出的记录,这个扩展段地址一直有效,直到读到下一个扩展段地址记录.

绝对地址与扩展段地址的关系如下：

绝对地址=数据记录中的地址[ZZZZ]+移位后的扩展段地址。