**https://ru.wikipedia.org/wiki/Intel\_HEX**

**Intel HEX** — формат [файла](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), предназначенного для представления произвольных двоичных данных в текстовом виде. По историческим причинам является стандартом де-факто при прошивке разнообразных микросхем с памятью (микроконтроллеров, ПЗУ, EEPROM и т. п.). Соответственно большинство инструментов подготовки образов прошивки (компиляторы, редакторы, просмотрщики и т. п.) умеют работать с этим форматом.

Файл обычно имеет расширение .hex. Названия I8HEX, I16HEX и I32HEX иногда используются для определения набора записей, используемых в файле. I8HEX использует только записи с типами 00/01 (16-битная адресация), в I16HEX добавляются записи 02/03 (20-битная адресация), и в I32HEX добавляются записи 04/05 (32-битная адресация).

Достоинством формата (в отличие от простого двоичного) является возможность указывать только определенные области адресов (с точностью до байта). Многие микроконтроллерные архитектуры имеют несколько областей программирования с обширными пустотами в адресации между ними.

**Содержание**

* [1 Формат записи](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.A4.D0.BE.D1.80.D0.BC.D0.B0.D1.82_.D0.B7.D0.B0.D0.BF.D0.B8.D1.81.D0.B8)
  + [1.1 Объяснение первое](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.9E.D0.B1.D1.8A.D1.8F.D1.81.D0.BD.D0.B5.D0.BD.D0.B8.D0.B5_.D0.BF.D0.B5.D1.80.D0.B2.D0.BE.D0.B5)
  + [1.2 Объяснение второе](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.9E.D0.B1.D1.8A.D1.8F.D1.81.D0.BD.D0.B5.D0.BD.D0.B8.D0.B5_.D0.B2.D1.82.D0.BE.D1.80.D0.BE.D0.B5)
  + [1.3 Запись адреса сегмента](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.97.D0.B0.D0.BF.D0.B8.D1.81.D1.8C_.D0.B0.D0.B4.D1.80.D0.B5.D1.81.D0.B0_.D1.81.D0.B5.D0.B3.D0.BC.D0.B5.D0.BD.D1.82.D0.B0)
  + [1.4 Запись расширенного адреса](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.97.D0.B0.D0.BF.D0.B8.D1.81.D1.8C_.D1.80.D0.B0.D1.81.D1.88.D0.B8.D1.80.D0.B5.D0.BD.D0.BD.D0.BE.D0.B3.D0.BE_.D0.B0.D0.B4.D1.80.D0.B5.D1.81.D0.B0)
  + [1.5 Запись конца файла](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.97.D0.B0.D0.BF.D0.B8.D1.81.D1.8C_.D0.BA.D0.BE.D0.BD.D1.86.D0.B0_.D1.84.D0.B0.D0.B9.D0.BB.D0.B0)
* [2 Примеры](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.9F.D1.80.D0.B8.D0.BC.D0.B5.D1.80.D1.8B)
  + [2.1 Пример 1](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.9F.D1.80.D0.B8.D0.BC.D0.B5.D1.80_1)
  + [2.2 Пример 2](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.9F.D1.80.D0.B8.D0.BC.D0.B5.D1.80_2)
  + [2.3 Пример 3](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.9F.D1.80.D0.B8.D0.BC.D0.B5.D1.80_3)
* [3 См. также](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.A1.D0.BC._.D1.82.D0.B0.D0.BA.D0.B6.D0.B5)
* [4 Ссылки](https://ru.wikipedia.org/wiki/Intel_HEX#.D0.A1.D1.81.D1.8B.D0.BB.D0.BA.D0.B8)

**Формат записи**

**Объяснение первое**

Файл состоит из текстовых ASCII строк. Каждая строка представляет собой одну запись. Каждая запись начинается с двоеточия (**:**), после которого идет набор [шестнадцатеричных цифр](https://ru.wikipedia.org/wiki/%D0%A8%D0%B5%D1%81%D1%82%D0%BD%D0%B0%D0%B4%D1%86%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F) кратных [байту](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82):

* Начало записи (**:**).
* Количество [байт](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82) данных, содержащихся в этой записи. Занимает один байт (две шестнадцатеричных цифры), что соответствует 0…255 в [десятичной системе](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%81%D1%8F%D1%82%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F).
* Начальный адрес блока записываемых данных — 2 байта. Этот адрес определяет абсолютное местоположение данных этой записи в двоичном файле.
* Один байт, обозначающий тип записи. Определены следующие типы записей:
  + 0 — запись содержит данные двоичного файла.
  + 1 — запись обозначает конец файла, данных не содержит. Имеет характерный вид «:00000001FF».
  + 2 — запись адреса сегмента (подробнее см.ниже).
  + 4 — запись расширенного адреса (подробнее см.ниже).
* Байты данных, которые требуется сохранить в EPROM (их число указывается в начале записи, от 0 до 255 байт).
* Последний байт в записи является [контрольной суммой](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D1%83%D0%BC%D0%BC%D0%B0). Рассчитывается так чтобы сумма всех байтов в записи была равна 0.
* Строка заканчивается стандартной парой CR/LF (0Dh 0Ah).

**Объяснение второе**

Файл формата Intel HEX может состоять из любого количества записей. Каждая запись представляет собой [ASCII](https://ru.wikipedia.org/wiki/ASCII)-строку состоящую из нескольких пар 16-ричных цифр. Строка должна начинаться с символа двоеточия и оканчиваться парой символов CR и LF. Запись состоит из пяти полей следующего формата:  
:LLAAAATTDD...CC  
Каждая группа букв (LL, AAAA, TT и т. д.) представляет собой отдельное поле. Каждая буква — отдельную 16-ричную цифру (4 бита). Каждое поле состоит, как минимум, из двух 16-ричных цифр (байт). Ниже представлена расшифровка полей записи:

* : Каждая запись в файле Intel HEX должна начинаться с двоеточия.
* LL Поле длины — показывает количество байт данных (DD) в записи.
* AAAA Поле адреса — представляет начальный адрес записи.
* TT Поле типа. Оно может принимать следующие значения:
  + 00 запись содержит данные двоичного файла.
  + 01 запись является концом файла.
  + 02 запись адреса сегмента (подробнее см. ниже).
  + 03 Start Segment Address Record.
  + 04 запись расширенного адреса (подробнее см. ниже).
  + 05 Start Linear Address Record.
* DD Поле данных. Запись может содержать несколько байт данных. Количество байт данных должно соответствовать полю LL.
* CC Поле контрольной суммы. Поле контрольной суммы вычисляется путём сложения значений всех байт (пар 16-ричных цифр) записи по модулю 256 с последующим переводом в [дополнительный формат](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4_%28%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%B0%29) (отнять получившееся значение от 0). Таким образом, если просуммировать все пары шестнадцатеричных чисел, включая LL, AA, TT, DD, CC, получится 0.

**Запись адреса сегмента**

Файл формата Intel HEХ может содержать записи для процессоров Intel 8086, которые определяют адрес сегмента (иначе говоря — номер параграфа, один параграф — это 16 байт). Если используется директива H167, то этот тип записи заменяет запись расширенного линейного адреса. Номер параграфа используется как смещение адреса для всех записей данных, следующих за этой записью. Смещение представляет собой сдвиг адреса на четыре бита влево. Например, смещение равное 0x1234 даст реальный адрес 0x12340. Пример записи:  
:020000021000EC где:

* 02 Количество байт данных
* 0000 В этом типе записи всегда равно 0000
* 02 Тип записи 02 (запись адреса сегмента).
* 1000 Номер параграфа (реальный адрес будет равен 0x10000).
* EC Контрольная сумма

**Запись расширенного адреса**

Когда используется директива H167, файл формата Intel HEX возможно будет содержать записи расширенного линейного адреса. Этот тип записи определяет значения двух старших байт абсолютного адреса (биты 16-31). Эти два старших байта адреса будут применяться как смещение адреса для всех записей данных, следующих за этой записью. Пример записи:  
: 0200000400FFFB  
где:

* 02 Количество байт данных
* 0000 Всегда равно 0
* 04 Тип записи 04 (запись расширенного линейного адреса)
* 00FF Старшее слово смещения адреса (0x00FF0000)
* FB Контрольная сумма

**Запись конца файла**

Файлы формата Intel HEX должны иметь запись Конца Файла (EOF).[[*источник не указан 1267 дней*](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8_%D0%BD%D0%B0_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8)] Пример записи:  
:00000001FF где:

* 00 Количество байт записи
* 0000 Поле игнорируется. Не имеет значения, что там записано, но обычно всегда равно 0
* 01 Тип записи 01 (Конец Файла)
* FF Контрольная сумма, вычисленная как 01h + NOT(00h + 00h + 00h + 01h).

**Примеры**

**Пример 1**

:10010000214601360121470136007EFE09D2190140

:100110002146017EB7C20001FF5F16002148011988

:10012000194E79234623965778239EDA3F01B2CAA7

:100130003F0156702B5E712B722B732146013421C7

:00000001FF

     Начало записи      Количество байт данных в этой записи (строке)      Адрес, по которому размещаются данные этой записи      Тип записи      Данные      Контрольная сумма записи

**Пример 2**

Пример одной записи данных:  
:10246200464C5549442050524F46494C4500464C33 где

* 10 Количество байт данных (16 байт)
* 2462 Адрес памяти, куда будет помещена запись.
* 00 Тип записи — данные.
* 464C...464C Данные
* 33 Контрольная сумма записи

**Пример 3**

:020000021000EC

:10C20000E0A5E6F6FDFFE0AEE00FE6FCFDFFE6FD93

:10C21000FFFFF6F50EFE4B66F2FA0CFEF2F40EFE90

:10C22000F04EF05FF06CF07DCA0050C2F086F097DF

:10C23000F04AF054BCF5204830592D02E018BB03F9

:020000020000FC

:04000000FA00000200

:00000001FF