



ARM Based Microcontroller

HEX Parser

Lecture 20



Advanced RISC Machines

Introduction

01

HEX Parser

HEX File

The HEX file is read by a programmer to write the machine code into a PROM or is transferred to the target system for loading and execution. a compiler or assembler converts a program's source code (such as in C or assembly language) to machine code and outputs it into a HEX file. Common file extensions used for the resulting files are (.HEX).

HEX file format:

HEX file consists of lines of ASCII text that are separated by line feed or carriage return characters or both. Each text line contains hexadecimal characters that encode multiple binary numbers. The binary numbers may represent data, memory addresses, or other values, depending on their position in the line and the type and length of the line. Each text line is called a record. Each text line is called a record.

HEX Record

A record (line of text) consists of six fields (parts) that appear in order from left to right:

1. **Start code:** one character, an ASCII colon ':'
2. **Byte count:** two hex digits (one hex digit pair), indicating the number of bytes (hex digit pairs) in the data field. The maximum byte count is 255 (0xFF). 16 (0x10) and 32 (0x20) are commonly used byte counts.
3. **Address:** four hex digits, representing the 16-bit beginning memory address offset of the data. The physical address of the data is computed by adding this offset to a previously established base address, thus allowing memory addressing beyond the 64 kilobyte limit of 16-bit addresses. The base address, which defaults to zero, can be changed by various types of records. Base addresses and address offsets are always expressed as big endian values.

HEX Record

- 4. **Record type:** two hex digits, 00 to 05, defining the meaning of the data field.
- 5. **Data:** a sequence of n bytes of data, represented by $2n$ hex digits. Some records omit this field (n equals zero). The meaning and interpretation of data bytes depends on the application.
- 6. **Checksum:** two hex digits, a computed value that can be used to verify the record has no errors.

examples of the a HEX file record:

:100050000C943F0011241FBECFE5D8E0DEBFCDBF1A The line must start with a colon, :, followed by the number of data bytes on the line, in this case 0x10 or 16 decimal. Each data byte is represented by 2 characters.

HEX Record

```
:0200000040800F2
:100000000050002021020008270200082B020008EF
:100010009D020008A1020008E902000800000000009B
:10002000000000000000000000000000000000003103000894
:10003000350300080000000000390300083D030008F4
:100040004D0300084D0300084D0300084D03000850
:100050004D0300084D0300084D0300084D03000840
:100060004D0300084D0300084D030008590800081F
:100070004D0300084D0300084D0300084D03000820
:100080004D0300084D0300084D0300084D03000810
:100090004D0300084D0300084D0300084D03000800
:1000A0004D0300084D0300084D0300084D030008F0
:1000B0004D0300084D0300084D0300084D030008E0
:1000C0004D0300084D0300084D0300084D030008D0
:1000D0004D0300084D0300084D0300084D030008C0
:1000E0004D0300084D0300084D030008000000000008
:1000F0000000000000000000000000000000000000
```

Checksum

Checksum Calculations

Each line must end with a checksum. This is the 2s-complement of the sum of the number of bytes, plus the address plus the data. To do this, add up the number of bytes, the address and all the data and discard any carry to give an 204 8-bit total.

Write this in binary, then invert each digit to give 1s-complement. Add one to give 2s-complement.

The checksum in this example line :**040010001122334442** is **42**.

This is calculated as follows:

1. $0x04 + 0x00 + 0x10 + 0x11 + 0x22 + 0x33 + 0x44 = 0xBE$
2. $0xBE$ is 10111110 in binary.
3. Invert this value e.g. by XOR with $0xFF$ gives 01000001 = $0x41$.
4. Add 1, gives the result $0x42$, which matches the checksum shown.

HEX Record

Record types

HEX files have six standard record types

1. Data record

Hex code: 00

- Example: : 0B0010 00 6164647265737320676170A7
- Description: Contains data and a 16-bit starting address for the data. The byte count specifies the number of data bytes in the record. The example shown next has 0B (eleven) data bytes (61, 64, 64, 72, 65, 73, 73, 20, 67, 61, 70) located at consecutive addresses beginning at address 0010.

2. End Of File record

Hex code: 01

- Example: 000000 01 FF
- Description: Must occur exactly once per file in the last line of the file. The data field is empty (thus byte count is 00) and the address field is typically 0000.

HEX Record

3. Extend Segment Address record

- Hex code: 02
- Example: 020000 02 1200EA
- Description: The data field contains a 16-bit segment base address (thus byte count is always 02) compatible with 80x86 real mode addressing. The address field (typically 0000) is ignored. The segment address from the most recent 02 record is multiplied by 16 and added to each subsequent data record address to form the physical starting address for the data. This allows addressing up to one megabyte of address space.

4. Start Segment Address record

- Hex code: 03
- Example: 040000 03 00003800C1

HEX Record

5. Extended Linear Address record

- Hex code: 04
- Example: 020000 04 FFFFC

6. Start Linear Address record

- Hex code: 05
- Example: 040000 05 000000CD2A



STM32
Is AWESOME

Session LAb

Create your own
HEX PARSER

*Time To
Code*



THANKS!

Do you have any questions?

www.imtschool.com

www.facebook.com/imaketechnologyschool/

This material is developed by IMTSchool for educational use only

All copyrights are reserved