Section 14

Appendix E - Object File Formats

14.1 Intel-HEX Format

The Intel® HEX object file format is supported by many assemblers, compilers, utilities and device programmers to specify executable byte code for embedded processors. An Intel HEX file is a 7-bit ASCII text file that contains a sequence of data records, address records and an end record. Every record is a line of text that starts with a colon and ends with CR and LF. Each record contains an 8-bit checksum to identify corrupted data.

Each line of an Intel HEX file consists of six fields:

1. **Start code**: one character, an ASCII colon ':'.
2. **Byte count**: two hex digits representing the number of bytes (hex digit pairs) in the data field, from 0–255. C51ASM outputs 16 bytes or fewer per record.
3. **Address**: four hex digits representing the 16-bit starting memory address for the data within a 64 KB block. The 64 KB limit is exceeded by specifying higher address bits via additional record types. The address is big endian (e.g. higher order byte first).
4. **Record type**: two hex digits, 00 to 05, defining the type of the data field.
5. **Data**: a sequence of *N* bytes of the data themselves (2*N* hex digits).
6. **Checksum**: two hex digits representing the two's complement modulo 256 sum of the values in all fields except fields 1 and 6 (Start code ":" and Checksum). It is calculated by adding together the hex-encoded bytes (hex digit pairs) as an 8-bit sum (leaving only the LSB of the result) and making a 2's complement. If the checksum is correctly calculated, adding all the bytes (the byte count, both bytes in address, the record type, each data byte and the checksum) together will always result in a value wherein the least significant byte is zero (0x00).

   For example, on :0300300002337A1E, calculate 03 + 00 + 30 + 00 + 02 + 33 + 7A = E2, and 2's complement is 1E

Table 14-1. Intel-Hex Record Types

Type	Name	Description
00	Data Record	Contains data and 16-bit offset address. The format is as described above.
01	End Of File Record	The file termination record. The data field is empty. Has to be the last line of the file, and only one per file permitted. The Intel specification allows the End Of File record to contain a start address for the program being loaded, e.g. :00AB2F0125 would make a jump to address AB2F. For C51ASM the address field is always zero, e.g. ':00000001FF'.
02	Extended Segment Address Record	One megabyte (20-bit) segment-base address. Used when the object file exceeds 64 KB. The address specified by the 02 record is multiplied by 16 (shifted 4 bits left) and added to the subsequent 00 record addresses. The address field of this record is 0000 and the byte count is 02 (the segment is 16-bit, big endian). In the Intel specification the least significant hex digit of the segment address is always 0. For C51ASM the segment is fixed on 64 KB boundaries making the three least significant hex digits always 0, for example ':0200000230009C' starts the segment at 0x30000.
03	Start Segment Address Record	Specifies the initial contents of the CS:IP registers for 80x86 processors, but is not used in C51ASM. The address field is 0000, the byte count is 04, the first two bytes are the CS value, and the latter two are the IP value.
04	Extended Linear Address Record	Specifies the upper address word for full 32-bit (4GB) addressing. The address field is 0000, the byte count is 02 and the two data bytes represent the upper 16 bits of the 32 bit address. C51ASM supports only up to 24-bit (16MB) addressing, so the higher order byte is always zero.
05	Start Linear Address Record	Specfies the value of the EIP register on 80386 and higher CPUs. The address field is 0000, the byte count is 04 and the 4 data bytes represent the 32-bit value loaded into the EIP register. C51ASM does not use this record type.

Example:

```
:10010000214601360121470136007EFE09D2190140
:100110002146017EB7C20001FF5F16002148011988
:10012000194E79234623965778239EDA3F01B2CAA7
:100130003F0156702B5E712B722B732146013421C7
:00000001FF
```

14.2 Motorola SREC Format

Motorola S-records (SREC) are an alternative to the Intel HEX files that are supported by some device programmers, especially for devices with memory above 64KB. An SREC format object file consists of a series of ASCII records. All hexadecimal numbers are big endian. The records have the following structure.

Each line of an SREC file consists of six fields:

1. **Start code**: one character, an ASCII 'S'.
2. **Record type**: one ASCII digit, 0–9, defining the type of the data field.
3. **Byte count**: two hex digits indicating the number of bytes (hex digit pairs) that follow in the rest of the record (in the address, data and checksum fields).