

```

//=====
=====
//TITLE:
//  S-Record 格式详解
//AUTHOR:
//  norains
//DATE:
//  Friday 25-April-2008
//Environment:
//  NULL
//=====
=====

```

一个完整的 MOTOROLA S-Record 格式数据包含如下区域:

<type> <length> <address> <data> <checksum>

各字段的意思分别如下:

<type>: 标示记录的类型, 该字段占据 1-byte。它可以有如下 数值: "S0", "S1", "S2", "S3", "S5", "S7", "S8", "S9"

"S0" -- 记录描述信息

"S1", "S2", "S3" -- 记录存储的数据。这三者的区别在于地址 (address) 的长度不同, S1 为 2-byte, S2 为 3-byte, 以及 S3 为 4-byte。

"S5" -- 包含了"S1", "S2", "S3"的信息。

"S7", "S8", "S9" -- 确定程序的开始地址。这三者的区别也在于地址 (address) 的长度不同, S9 为 2-byte, S8 为 3-byte, 以及 S7 为 4-byte。

<length>: 标示了数据的长度, 确切地说, 是 <address>, <data> 和<checksum>这三个字段的 byte 的个数。该字段占据 1-byte。

<address>: 标示了数据写入的起始地址。该字段的长度取决于<type>的取值。

<data>: 标示了存储的数据。该字段占据的 byte 个数可以这样计算: <length>的值

- <address>字段的长度（取值为 2、3、4） - 1（<checksum>字段的长度）

<checksum>：标示校验位，占据 1-byte。该数据可以由<address>和<data>的数据累加然后每 bit 取反获得。

最后我们以一条数据记录格式作为例子：

S30A801000930300000000CF

S3 是记录的格式（<type>）。

0A 是长度（<length>），为十六进制，标识有 10 个 byte。因为两个十六进制才是 1-byte，所以 801000930300000000CF 的长度为 10 而不是 20。换句话说，这数据列应该如此表示：80 10 00 93 03 00 00 00 00 CF。

因为 type 是 S3，所以地址（<address>）占 4-byte，因此需要写入的地址为：0x80100093。

既然已经知道了地址，那么数据（<data>）很显然就是 0x0300000000 了。

最后的 1-byte 就是校验位（<chksum>）。例子数值由此得来：0xFF - ((0x0A + 0x80 + 0x10 + 0x00 + 0x93 + 0x03 + 0x00 + 0x00 + 0x00 + 0x00) & 0xFF)