常见嵌入式系统可执行文件格式解析

嵌入式系统编程文件格式多种多样。为方便嵌入式系统开发和深度理解各种目标系统，论文详细分析了多种主流的嵌入式可执行文件(即机器码文件)格式。比较了不同格式的异同点，并介绍了各种嵌入式文件格式的主要硬件系统及目标器件。

1 常见文件格式解析

虽然不同的开发集成环境和不同的硬件架构使得嵌入式设备中可执行文件的格式不尽相同，但基本上包含以下一些典型特征：

① 可执行文件的基本信息，如文件大小、时间、权限等。

② 与硬件架构相关的二进制代码和数据。

③ 符号表与符号重定位表。

从文件本身所包含的信息来看，嵌入式系统可执行文件主要有：纯数据类文件，记录类文件以及描述类文件。

1.1 纯数据文件格式

纯数据文件就是指文件中不包含地址和校验信息等，只包含纯粹的用户数据。纯数据文件主要包括两种：BIN 格式和 HEX 格式。BIN 格式即二进制的文件，使用时每 8 位作为一个字节来用，传输 BIN 文件时并不进行 ASCII 编码。BIN 文件格式是没有结束标记的。显然，BIN 文件格式是除了一些压缩格式(如 ZIP 和 RAR)之外最节省空间的格式，它主要是用来编程存储器器件的，通常会建议将 EPROM 数据存储为 BIN 格式以作为标准文件格式来下载[2].HEX 格式相比 BIN 格式的区别就是，每个字节都转成了 2 个 0~9 和 A~F 之间值的 ASCII 码。整个文件是分行表示的，每行一般是 16 或 32 个字节，行以空格加回车表示结束。

1.2 记录类文件格式

记录类文件中所有的数据行都叫做记录，每一条记录一般都分为多个域来清晰地说明记录的类型、记录中数据的数量、数据的存储地址、数据校验信息等。而且，记录类文件一般都有关于该文件开始和结束的标记信息。各种记录类文件格式的差异就体现在不同公司定义的记录标记以及一些域格式和顺序的差别上。记录类文件常见的就是各种 MCU 的烧录文件。

1.2.1 Intel HEX 文件格式（.HEX）

Intel HEX 是一种较老的文件格式。它是用 ASCII 字符来表示文件中数据域的二进制数据。文件中每一行是一条 HEX 记录，由一个回车和一个换行结束。

Intel HEX 文件经常被用于将程序或数据传输存储到 ROM、EPROM,大多数编程器和模拟器使用 Intel HEX 文件。

记录格式如下：

:BBAAAATTHHHH…HHHCC

:--记录开始标记。

BB--长度域，占一个字节。代表这条记录中数据的字节数，每条记录最大数据字节数是 255 字节，通常为 32 字节。长度域只计数数据域的大小。

AAAA--地址域，占两个字节，存储方式是 MSB.地址域指出这条记录的第一个数据字节的地址，随后地址逐一增加以存储下面的数据。整个文件中各记录的地址顺序是不重要的，文件中有时也会有一段空的地址。

TT--类型域，1 字节。表示该条记录的类型。

HH--数据域，长度受到记录类型的限制。数据记录以字节为单位。

CC--校验和，占一个字节。计算方法是将本记录中除了冒号和校验字节之外的所有字节加起来，将模除 256 得到的余数取补码。

Intel HEX 文件格式的 6 种记录说明如表 1 所列。

记录类型	长度域	地址域	类型域	数据域	校验和
数据记录	1 字节	2 字节	00	最大 255 字节	00～FF
结束记录	00	0000	01	无	FF
扩展段地址记录	02	0000	02	2 字节的偏移地址	00～FF
开始段地址记录	04	0000	03	4 字节,前 2 个字节是下载到代码段的数据,后 2 个字节是下载到 IP 寄存器的数据	00～FF
扩展线性地址记录	02	0000	04	2 字节扩展线性地址	00～FF
开始线性地址记录	04	0000	05	指定 Intel 处理机的 4 字节开始地址,存在处理机的 EIP 寄存器	00～FF

表 1　Intel HEX 文件格式的记录类型

1.2.2　Motorola S/EXORciser/EXORmacs/32 位格式

Motorola 公司的产品主要基于 Motorola S 记录格式，这种文件格式以一种可打印的格式编码程序和数据文件来实现计算机系统间的传输，可以使下载文件

实现可视化编辑。S 记录由特殊排列的一串 ASCII 字符序列组成，最大长度为 78 字节。一个文件内的 S 记录一般没有特定的顺序要求。目前主要有 3 种 S 记录文件格式：S19 格式，地址域是 16 位，最大可写 64 Kb 的文件；S28 格式，地址域 24 位，最大可写 16 Mb 的文件；S37 格式，地址域 32 位，最大可写 4 Gb 的文件。

S 记录的一般格式是：类型域--长度域--地址域--数据域--校验和。S 记录的各种记录类型有不同的格式和意义，具体如表 2 所列。

Motorola S19（又称 Motorola EXORciser）格式是以 S0 记录开始，数据记录都是以 S1 标识的 2 字节地址类型的记录，S9 标识结束记录格式。Motorola S28（又称 Motorola EXORmacs）格式是以 S0 记录开始，数据记录都是以 S2 标识的 3 字节地址类型的记录，S8 标识结束记录格式。Motorola S37（又称 Motorola 32 位）格式是以 S0 记录开始，数据记录以 S3 字符标识的 4 字节地址类型的记录，S7 标识的结束记录格式。

类型域/数据范围	长度域	地址域	数据域	检验和
S0 0x5330	03～FF	0000	≤32 个字节	1 个字节
S1 0x5331	03～FF	2 字节	≤32 个字节	
S2 0x5332	03～FF	3 字节	≤32 个字节	
S3 0x5333	03～FF	4 字节	≤32 个字节	
S5 0x5335	03	记录前 S1 S2 和 S3 的记录个数	无	
S7 0x5337	03～FF	4 字节的开始执行地址	≤32 个字节	
S8 0x5338	03～FF	3 字节的开始执行地址	≤32 个字节	
S9 0x5339	03	2 字节的开始执行地址	无	

表 2　S 记录的记录类型

1.2.3　Tektronix HEX/扩展的 Tektronix HEX 格式

Tektronix HEX 格式如下所示：

/AAAABBEEDDDDDDDDCC

/:记录开始标记符。

AAAA:地址域。占两个字节，记录了第一个数据字节的地址。

BB:长度域。占 1 个字节，因此每条记录的最大数据个数为 255 字节。

EE:地址和长度数据的校验和域。大小为 1 个字节。

DD:数据域。最大为 255 个数据字节，但一般记录中为 32 个字节。

CC:数据的校验和域。

文件的结束记录中包括记录开始符（"/"）、传输地址、数值为"00"的长度以及传输地址和长度的校验和字节。一种可选择的中断记录以 2 个开始符("//")为记录开始，接着是一串任意的 ASCII 字符串。

扩展的 Tektronix HEX 相对 Tektronix HEX 格式具有较为灵活的记录类型和地址空间。它包括 3 种类型的记录：数据、符号和结束记录。数据记录中的是目标代码；符号记录中的是编程部分的信息，编程器下载时符号记录是被忽略的；结束记录包括头部、下载地址和目标代码。

扩展的 Tektronix HEX 格式模型如下：

%BBMEEAAAAADDDDDD---DCC

%:占 1 个 ASCII 字符，标示扩展 Tek HEX 记录开始。

BB:长度域。占 1 个字节，表示该记录中除开始标志之外的字节数。

M:类型域，一个 ASCII 字符。类型域中：6 代表数据记录，3 代表符号记录，8 代表结束记录。

EE:地址和长度数据的校验和，大小为 1 个字节。

AAAAA:地址域。地址域包括 2~17 个字符，第 1 个字符决定了地址的长度，一般为 8,0 代表 16 个字符的地址长度。地址域第 1 个字符之外的字符指出了数据在存储器存放的地址。

DD:数据域。包括可执行代码，下载到存储器的数据以及传输中的描述信息。

CC:校验和，占一个字节。校验和域对本条记录中除了"%"和校验数据本身之外的所有数据计算校验和。

根据是否有数据和结束记录，记录中域的数量是不一样的，2 种记录都有 6 个字符的头以及 2~17 个字符的地址。

1.3 描述类文件格式

描述类文件格式的文件信息有相对丰富的文件格式，文件不仅有关于数据的记录信息，还包含了关于目标机的一些说明信息，改变硬件结构的信息或者测试信息等。该类型的文件最常见的是用于 PLD 和 FPGA 的下载文件。

1.3.1 JEDEC 文件

JEDEC（Joint Electron Device Engineering Council）标准是用于现场可编程逻辑器件的。该标准定义了在开发系统和编程器之间传输熔丝状态的数据格式。它并不涉及诸如逻辑阵列类型和输出宏单元等设备结构信息。现场可编程逻辑器件比可编程的存储器件需要更多的测试信息，所以该标准还定义了一些简单的测试格式。JEDEC 格式的合法字符包括所有可打印的 ASCII 字符和 4 个控制字符，包括 STX（Start of Text）、ETX（End of Text）、CR（Carriage Return）和 LF（Line Feed）。一般使用 BNF（BackusNaur Form,巴科斯诺尔范式）来定义 JEDEC 格式的语法。

JEDEC 格式的第 1 个域是强制的关于设备信息设计规格书，这个域没有开始标记符。在 JEDEC 文件中每一个域都有域识别符以指明该域的类型，每一个域都是以星号结束。JEDEC 文件各种域识别符如表 3 所列。

描述及定义域主要用来说明用户和厂商信息、日期和版本号等各种信息。

N--说明域。用来在 JEDEC 文件传输中插入说明信息，编程器是忽略这一域的。

D--器件域。编程器支持的器件并不是遵从这一域，而是由 QF 和 QP 域以及手动选择器件而决定。

QF,QP 和 QV--数值域。QF 子域定义了器件中熔丝的数量，在器件编程或者测试域之前，这些数值域都是必须的；只有测试域的文件不需要 QF 域，编程数据的文件不需要 QP 和 QV 域;QF 子域向编程器指出了熔丝数据需要多少存储器，多少熔丝要被设置到初始条件以及计算校验和所包含的熔丝数；QP 子域定义了测试向量中的引脚数目或者测试条件；QV 子域指出了测试向量的最大数。

器件编程域。每一个熔丝有 2 种可能的状态：0 为低阻连接，1 为高阻连接。熔丝编号从 0 开始一直到最大熔丝数目。L 域中 L 之后是该域第一个熔丝的十进制序号数字，接下来就是指示熔丝状态的二进制数字列表，熔丝号和熔丝列表之间由空格或者回车分开。F 域定义了 L 域中未明确定义的熔丝状态信息。C 域是把从第一个熔丝开始的整个器件的熔丝状态每 8 位作为一个字节来计算的，最后一个字中若有未使用的位则填 0.U 域允许指定 JEDEC 文件中的不影响逻辑和电特性的用户数据熔丝。E 域允许定义不影响器件逻辑功能也不影响已有的 JEDEC 文件的特性，数据可以用二进制或者十六进制来表示。J--器件标识域。提供逻辑识别码指出文件对应器件的逻辑结构，由以空格分开的分别大于等于 0 的 2 个十进制数组成：第 1 个是结构码，第 2 个是引脚码。J 域中的编码是由 JEDEC 组织分配的。

设备测试域。功能性的测试由测试向量包含的测试条件来定义，每一个测试向量包括 n 个测试条件。n 是器件引脚数，测试条件如表 4 所列。

X--默认测试条件域。定义了未精确定义的测试向量的输入逻辑电平，即把 QV 域中的测试向量 1 设置为默认输入测试条件。若使用 X 域，应当在 QV 和 QP 域之后第一个测试向量之前。V--测试向量域。由十进制的向量编号开始，之后是空格以及每个引脚的一串测试条件。P--Pin 引脚序列域。P 域映射出了器件输出引脚，一般和 V 域一起使用。代码如下所示：

```
P 1 2 3 4 5 6 14 15 16 17 7 8 9 10 11 12 13 18 19 20 *

V0001 111000HLHHNNNNNNNNNN*

V0002 100000HHHLNNNNNNNNNN*
```

编程及测试选项类域：G 为熔丝安全域。用来使能某些逻辑器件的加密熔丝，在 G 域发送 1，即 G1*；R、S、T 为属性分析测试；A 为访问时间域，定义了测试向量在十亿分之一秒内的传播延时。

识别符	(n.a.)	A	C	D	E	F	G
含义	设计说明(域)	访问时间	(熔丝)校验和	器件类型(废弃的)	电子数据	默认熔丝状态	保险熔丝
识别符	J	L	N	P	QF	QP	QV
含义	器件标识	熔丝列表	注释	(器件)引脚序列	器件中熔丝数目	器件封装引脚数目	测试向量最大数目
识别符	R	S	T	U	V	X	其余
含义	测试结果向量	测试开始向量	测试周期	用户数据	测试向量(域)	默认测试条件	留作以后使用

注：*——编程器必须识别的域；**——测试器必须识别的域；***——开发系统必须提供的信息

表 3　JEDEC 文件各种域识别符（点击查看大图）

测试条件	0	1	2～9	B	C	D
含义	驱动输入低	驱动输入高	驱动输入到高压	隐藏注册预载	驱动输入低 高 低	驱动输入低，快回转
测试条件	F	H	K	L	N	P
含义	浮数输入输出	测试输出高	驱动输入高 低 高	测试输出低	电源和输出引脚不测试	预载寄存器
测试条件	R	T	U	X	Z	其余字母
含义	读并保持寄存器状态	寄存器观测	驱动输入高，快回转	输出不测，输入取默认	高阻时测输入输出	保留

注：C K U 和 D 驱动信号只能在其他输入稳定后才能有效，L H 和 Z 测试只能在所有输入包括 C K U 和 D 稳定之后才能执行

表 4　设备引脚的各种测试条件(点击查看大图)

1.3.2　POF 文件

POF 文件中的包类型如表 5 所列。POF（Programmer Object File，编程目标文件） 格式提供高度压缩的数据文件，可方便有效地传输大量的逻辑器件编程信息。这种文件目前主要应用于 MAX 系列器件。POF 文件由文件头和各种包组成，每个包对应一个标记，指出包中的数据类型以及数据本身结构，各个包的长度和结构各异。每个数据包的最前 6 个字节是包类型和其余字节的数量。编程设备在读取 POF 文件时，若包类型未被识别，则该包弃之不用。在一个 POF 文件中，

除了结束标记包外，各种类型都可多次出现。有些包编程器是需要包中信息的，有些包编程器是不使用的（跳过），有些包则是编程器只读包中信息但不直接应用该信息的（只读）。

包类型	含　义	说　明
1	创建 ID 包	为产生 POF 文件的程序创建一个版本 ID 号
2	设备名包	要编程的目标设备的 ASCII 码名字
3	文本注释包	关于该 POF 的文本描述
4	xx	跳过
5	安全信息包	指出了目标设备是否可以被加密
6	逻辑地址和数据包	(只读)定义目标设备的一组逻辑地址以及相应该地址的逻辑数据，
7	电子地址和数据包	定义目标设备的一组电子地址及相应该地址的逻辑数据，从最低位地址的数据开始纵横向顺序排列，之后是行地址增加排列数据直到第一列满，之后就增加列地址，如此反复
8	结束包	指示 POF 文件结束，CRC 校验域是文件中除 CRC 之外的所有字节的 CRC 校验和　若 CRC 值为 0，则 CRC 校验值被忽略
9	符号表	跳过
10	测试向量包	在编程的测试阶段使 POF 文件包含有测试向量
12	电子地址和常数	跳过
14	可编程元素包	(只读)定义了目标设备中可编程的元素数量
17	逻辑地址和 32 位数据包	(只读)定义了目标设备的一组逻辑地址以及相应地址的逻辑数据，这些地址区域从低地址端开始向高地址延伸，开始地址和地址数量都是 4 字节的

表 5　POF 文件中的包类型

2　总结

为方便用户使用，目前大多数主流公司生产的编程器都能够在下载文件时自动进行某几种文件格式的检测识别以及文件格式的转换。然而这毕竟解决不了文件格式之间互不兼容的实际问题。因此，我们期望在未来几年内文件格式可以逐步走向统一，拥有一种或少数几种格式标准。