

吴华灯, 卢子晋, 叶世山. Altus 记录器事件文件格式分析及解码的实现[J]. 华南地震, 2016, 36 (2): 20-24. [WU Huadeng, LU Zijin, YE Shishan. Analysis and Decoding of Altus Recorder Event File Format[J]. South china journal of seismology, 2016, 36(2): 20-24.]

## Altus 记录器事件文件格式分析及解码的实现

吴华灯<sup>1, 2, 3</sup>, 卢子晋<sup>1, 2, 3</sup>, 叶世山<sup>1, 2, 3</sup>

(1. 广东省地震局, 广州 510070; 2. 中国地震局地震监测与减灾技术重点实验室, 广州 510070;  
3. 广东省地震预警与重大工程安全诊断重点实验室, 广州 510070)

**摘要:** 介绍了美国 Kinometrics 公司 Altus 系列记录器的事件文件格式, 从解码流程和解码的程序实现两个方面阐述了事件文件解码的实现过程, 最后对解码数据进行了验证。验证结果表明: 解码数据是正确的, 将为后续自动快速产出强震动参数、数据研究应用、数据格式统一等相关工作提供可靠的数据源。

**关键词:** Altus; 强震动; 记录器; 事件文件; 格式; 解码

中图分类号: P315.723

文献标志码: A

文章编号: 1001-8662 (2016) 02-0020-05

DOI: 10.13512/j.hndz.2016.02.004

## Analysis and Decoding of Altus Recorder Event File Format

WU Huadeng<sup>1, 2, 3</sup>, LU Zijin<sup>1, 2, 3</sup>, YE Shishan<sup>1, 2, 3</sup>

(1. Earthquake Administration of Guangdong Province, Guangzhou 510070, China;

2. Key Laboratory of Earthquake Monitoring and Disaster Mitigation Technology, Guangzhou 510070, China;

3. Key Laboratory of Guang Dong Province, Earthquake Early Warning and Safety Diagnosis of Major Projects, Guangzhou 510070, China)

**Abstract:** The paper introduces the event file format of kinometrics Altus series recorder, and describes the realization process of the event file decoding in two aspects from decoding flow and decoding procedure. The verification results show that the decoded data is accurate, which could provide reliable data source for subsequent automatic and fast produce strong motion parameters, data research and application, unified data format such as related work.

**Keywords:** Altus; Strong motion; Recorder; Event file; Format; Decoding

收稿日期: 2015-06-30

基金项目: 中国地震局地震科技星火计划项目(XH14038Y)

作者简介: 吴华灯(1980-), 男, 高级工程师, 主要从事地震观测研究和软硬件开发工作。

E-mail: gdea\_whd@aliyun.com.

0 引言

我国“十五”期间建成的强震动观测台网，大量使用了从美国 Kinemetrics 公司进口的 ETNA、K2 强震动记录器，将近 1 500 套，它们均属于 Altus 系列。目前，我国强震动观测台网强震动记录器种类较多、事件文件格式各异现状，给数据处理及研究应用带来诸多不便。考虑到作者开发的“国家强震动台网数据传输及监控管理软件”在第三版中要实现自动快速产出强震动参数以及在数据研究应用上更加便捷、高效的需要，作者在陆续实现其它强震动记录器事件文件解码的基础上，尚需继续对 Altus 系列记录器的事件文件进行解码<sup>[1-2]</sup>，以期获取可用的数据源，实现强震动参数的快速产出、强震动事件格式的统一等等。本文就 Altus 系列记录器的事件文件格式展开分析并讲解其解码的实现过程。

1 Altus 事件文件格式

1.1 Altus 系列记录器简要介绍

Altus 系列记录器由美国 Kinemetrics 公司生产，包括 Etna、K2、Mt.Whitney、Makalu、SMARTS、QDR、SSA2EVT 等系列型号。目前在我国强震动观测台网中使用的 Altus 系列主要有 Etna 和 K2 两款，均是早期性价比较高的强震动记录器。Etna 是 K2 的升级版本，K2 在 2007 年后停产，Etna 则在 2015 年后停产。自 2008 年中国数字化强震动台网正式运行以来，这两款记录器在我国的多次大地震中记录了大批量宝贵的数字化强震动记录。

1.2 Altus 事件文件格式

Altus 系列记录器的事件文件格式的主体结构是一样的，仅仅在文件头结构 FILE HEADER 上存在细微的差别。主体结构由四部分组成，分别是标签 TAG、文件头 FILE HEADER、帧头 FRAME HEADER 和数据区，如图 1 所示。标签 TAG 总是出现在文件头和每个帧头之前，即它是寻找文件头和帧头的标志，其大小固定为 16 字节。所以，EVT 文件的开始就是占 16 个字节空间的标签 TAG，第 17 个字节起就是文件头，文件头之后又是一个标签 TAG，显然其后就是帧头 FRAME HEADER，最后是数据区。文件头 FILE HEADER 因记录器的型号不同而存在着微小的差别。以

ETNA、K2 和 Mt.Whitney 为例，ETNA、K2 记录器的 FILE HEADER 是 2 040 字节，而 Mt.Whitney 记录器的是 2 736 字节。下面分别介绍各个结构的细节。



图 1 Altus 事件文件格式

Fig.1 Altus event file format

1.2.1 TAG 结构

TAG 结构主要由同步字符、字节序、文件格式版本、仪器类型代码、结构类型代码、文件头长度、数据长度、仪器系列号、校验和等九个成员组成，大小为 16 个字节，如表 1 所示。

表 1 TAG 结构  
Table 1 The TAG structure

成员变量	字节大小	备注
sync	1byte	同步字符
byteOrder	1byte	字节序
version	1byte	文件格式版本
instrumentType	1byte	仪器类型代码
type	4byte	结构类型代码
length	2byte	文件头长度
dataLength	2byte	数据长度
id	2byte	仪器系列号
checksum	2byte	校验和

1.2.2 FILE HEADER 结构

Altus 的头文件比较复杂，主要由只读参数 roParms 和读写参数 rwParms 两个大结构组成。只读参数 roParms 结构包括了仪器代码 instrumentCode、文件头版本号 headerVersion、文

件头大小 headerBytes 三个自变量和杂项参数、GPS 时钟信息、通道记录信息 3 个结构变量。rwParms 结构包括了杂项参数、GPS 时钟信息、通道记录信息、流信息和 modem 信息 5 个结构变量，结构组成如表 2 所示。需要注意的是，文件头版本号 headerVersion 主要为 K2 和 Mt.Whitney 定义，K2 有 100、110、130、140 四个版本号，Mt. Whitney 有 120、150 两个版本号，这两款仪器的文件头类型由 headerVersion 决定，而 Etna、Makalu、SMARTS、SSA2EVT 的文件头类型则由仪器代码 instrumentCode 决定。Makalu 对应的 instrumentCode 是 10，Etna2 的是 20，SMARTS 的是 30，SSA2EVT 的是 40。

表 2 文件头结构  
Table 2 The FILE HEADER structure

结构变量	成员变量	备注
roParms	id	3byte
	instrumentCode	1byte
	headerVersion	2byte
	headerBytes	2byte
	misc	结构变量
	timing	结构变量
rwParms	channel	结构变量
	misc	结构变量
	timing	结构变量
	channel	结构变量
	stream	结构变量
	modem	结构变量

1.2.3 FRAME HEADER 结构

FRAME HEADER 的大小为 32 字节，第一个帧头位于文件头之后的 TAG 之后，其余的帧头在后面每个 TAG 之后，由帧类型、仪器代码、记录器 ID、帧大小、块时间、通道位图、流参数、帧状态、帧状态 2、微秒、通道位图 1 和时间代码等 12 项组成，如表 3 所示。

1.2.4 数据区

每一组 TAG 和帧头后就是每一帧的数据区。数据区的数据归属哪个通道，取决于通道位图 channelBitMap 和 channelBitMap1，bit0 代表通道 1，bit1 代表通道 2，bit2 代表通道 3，最多到 24 通道，数据区的大小可由帧头的 frameSize 减去 32 得到。此外，可由 frameStatus 求得每个采样点的字节数，从而计算出每个扫描区的大小以及每一帧包含的扫描区数。数据的时间标识由块时间

表 3 帧头结构 Table 3 The FRAME HEADER structure		
成员变量	字节大小	备注
frameType	1byte	帧类型
instrumentCode	1byte	仪器代码
recorderID	2byte	记录器 ID
frameSize	2byte	帧大小
blockTime	4byte	块时间
channelBitMap	2byte	通道位图
streamPar	2byte	流参数
frameStatus	1byte	帧状态
frameStatus2	1byte	帧状态 2
msec	2byte	微秒
channelBitMap1	1byte	通道位图 1
timeCode	13byte	时间代码

blockTime 和 msec 共同决定。

2 事件文件解码的实现

2.1 解码流程

Altus 事件文件的解码过程并不复杂，但比较繁琐。根据图 1Altus 事件文件格式，第一步是读取文件头，首先读取文件的第一个 TAG，即文件头前面的 TAG。然后根据 TAG 的数据长度 dataLength 字段和文件头长度 length 字段判断是否是有效的文件头 TAG。如果是有效的文件头 TAG，则读入文件头的内容，确定文件头的版本类型，提取文件头的各项信息。第二步是读取帧头，同样先读取帧头前的 TAG，出于严谨，需要对同步字符 sync 作一个判断，如果同步字符不等于‘K’，则文件仅仅包含一个文件头，将不进行后面的任何处理。否则是一个带有数据的事件文件，需要从通道位图 channelBitMap 中读取通道数。最后一步是读数据区的数据，方法是逐个数据帧及数据帧中的每个扫描区的通道数据循环读取，当累计的扫描数大于数据总扫描数后结束数据读取，解码流程如图2 所示。

2.2 解码的程序实现

在程序设计时，首先需要对 TAG、FILE HERDER、FRAME HERDER 三个结构和相关常量进行定义，然后根据图 2 的解码流程，编写程序实现数据的解码。由于程序代码较长，受篇幅所限，仅将解码的核心过程列举如下：

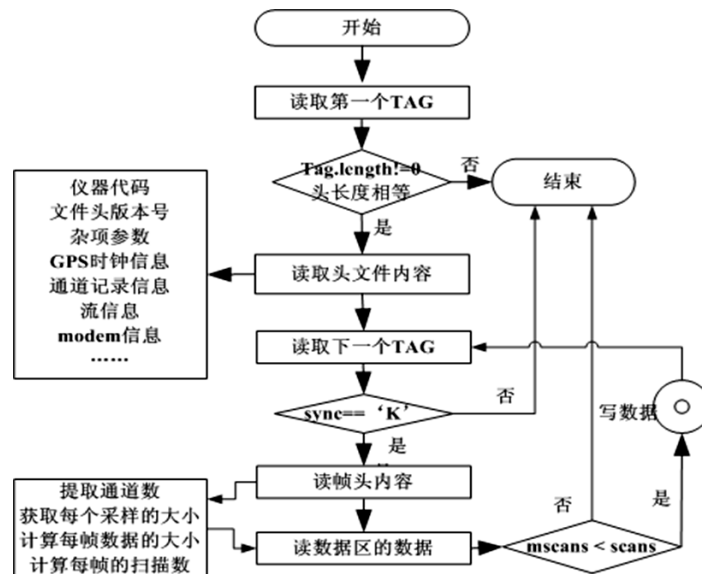


图2 解码流程

Fig.2 Decode flow

## (1) 读第一个 TAG。

```

infile = fopen (EvtName, "rb");
if (infile == NULL)
{
    return;
}
fread(&tag, sizeof(tag), 1, infile); //读第一个 TAG
if ( ( tag.dataLength! =0) || ( ( tag.length! =sizeof
(head)))) //判断数据长度及文件头长度
{
    return;
}

```

## (2) 读头文件的内容。

```

fread (&head, 10, 1, infile); //读进头文件内容
switch (headtype) //文件头类型匹配
{
    .....
}
//以下是按照匹配的文件头类型并根据文件头的结构
//逐个提取文件头的信息
instcode = head.roParms.instrumentCode; //提取仪器
//代码
headver = head.roParms.headerVersion; //提取文件
//头版本号
.....

```

## (3) 读帧头的内容。

```

fread (&tag, sizeof (tag), 1, infile); // 读第二个

```

## TAG

if (tag.sync != 'K') //如果没有找到特征字符 'K',  
程序结束, 否则继续执行后面的程序

```

{
    return;
}
fread (&frmhdr, sizeof (frmhdr), 1, infile); //读进
//帧头内容
bitmapinfo = frmhdr.channelBitMap; //提取通道信息
for (j = 0; j < MAX_CHANNELS; j++) { //计算
//通道数
tempchan = bitmapinfo & chanmask;
if (tempchan != 0) {
    chanct = chanct + 1;
}
chanmask = chanmask << 1;
}
framestat = frmhdr.frameStatus & 0xC0;
switch (framestat) //获取每个采样的大小
{
    case 0x40: bytesPerSample = 2; break;
    case 0x80: bytesPerSample = 3; break;
    case 0xC0: bytesPerSample = 4;
}
bytesPerScan = bytesPerSample * chanct; //计算每
//个扫描所占的字节数
frameSize = frmhdr.frameSize - 32; //计算每帧数据

```

的大小

`scansPerFrame = frameSize / bytesPerScan;` //计算每帧的扫描数

(4) 解码数据区的数据

```
i = fread (&data, frameSize, 1, infile);
for (i = 0; i < scansPerFrame; i++) {
for (j = 0; j < chanct; j++) {
//取出一个扫描中每通道的数据
}
mscans++;
}
```

### 3 解码数据的验证

对于每一种记录器事件文件解码的结果,作者都会进行严格的数据验证,本文也不例外。主要采用了数据文件的比较法。

数据文件的比较法是数值比较,即把厂家上位机软件生成的数据文件与本文解码出的数据文件进行比较,只要有一个值不相等,均视为不正确,因此能保证解码数据的正确性。如图3所示,两个文件比较的结果 0:0 字节差异、532249 字节全部匹配。



图3 文件比较

Fig.3 File comparison

### 4 结语

文章循序渐进地讲解了 Altus 记录器的事件文件格式、解码流程及解码的程序实现,并验证了解码数据的正确性,作为又一系列强震动记录器事件文件解码的实现的阶段成果,将为后续自动快速产出强震动参数、数据研究应用、数据格式统一等相关工作提供可靠的数据源。

### 参考文献:

- [1] 吴华灯, 闫俊义, 叶春明. Altus 强震数据采集器实时数据流解码的实现[J]. 华南地震, 2010, 30 (S1): 1-2.
- [2] 吴华灯, 叶春明. TDE-324CI 数据采集器强震实时数据流格式及解码过程浅析[J]. 防灾科技学院学报, 2011, 13 (4): 61-66.