Ｋ５データ修復ソフト(data_recov)説明書

Ver. 2006-09-21

T.Kondo

プログラム名　data_recov

機能

機能１：ヘッダーデータが不具合等で datachk においてヘッダーが検出できない K5/VSSP データファイルのヘッダー部分の修復を行います。

機能２：途中のヘッダーが不具合になっている K5/VSSP データファイルのヘッダー部分の修復を行います。

原理（機能１）：ヘッダーデータの破損具合を調査したところ、破損したヘッダーデータには以下に示される共通パターンが見られることが判明しました（ただし k51a で取得したデータのみの調査）。

正常ヘッダー：　FFFFFFFF 8Bxxxxxx　(x の部分はモード、時刻で変化)
異常ヘッダー：　FFFFFFxx 8BxxxxFF　(x の部分は変化)
（注：ヘッダー６４ビットを３２ビット整数×２の１６進数で表示）

そこで、"FFFFFF?? 8B????FF"(?は任意)のパターンをサーチすることにより、ヘッダー位置を検出します。(データによってはファイルの最初がヘッダーでない場合もあり、この場合に上記パターンのサーチにより最初のヘッダー位置を認識します。）なお、破損したヘッダーから時刻情報の復元は不可能であったため、最初のヘッダーにおける時刻はユーザーが与えなくてはなりません。

使用上の注意：

１．サンプリングパラメータ、データ開始時刻はユーザーが与える必要があります。

２．ヘッダー部分が修復されたファイルはデータ部分にも若干の不具合が残っているようです。実際の相関処理の例では遅延方向に山がスプリットしましたが、一番高いピークは正常のようです。ただしＳＮＲは劣化しています。

原理（機能２）：１秒間のデータは正常（データの欠損も超過もない）でヘッダー部分だけが異常であると仮定して、ヘッダー部分を正常なデータに置きかえる

実行方法

data_recov K5file [sfreq adbit numch start_time [HDpattern1 HDpattern2]]

ここで　K5file　----- ヘッダーを修復する K5 ファイル名

以下のパラメータは機能１による修復の場合に指定する
sfreq　----- サンプリング周波数 (MHz)
0.04,0.1,0.2,0.5,1,2,4,8,16,32,64
adbit　----- ＡＤビット数 (1,2,4,8)

numch　　----- チャンネル数 (1,4)
start_time --- データ開始時刻（以下のどちらかのフォーマットで）
　　"HH:MM:SS" または　"seconds in day"

以下のパラメータは異常ヘッダーパターンをデフォルト以外に設定
するときに指定する
HDpattern1 --- 異常ヘッダーパターン１　８桁１６進アスキー
　　'?'はその部分は何でもいいことを示す
　　（例：FFFF????）（デフォルトはFFFF????）
HDpattern2 --- 異常ヘッダーパターン２　８桁１６進アスキー
　　'?'はその部分は何でもいいことを示す
　　（例：8B????FF）（デフォルトは8B????FF）

備考：HDpattern1,HDpattern2は特定のマシン（k51a）で見られたパターン以外に対応するためのパラメータ

ヘッダーが修復されたデータ出力ファイル名は　K5 ファイル名＋".edit"となります。

実行例　（機能１）

```
data_recov F:¥0202151500a.dat 4 1 4 15:15:01

************ data_recov  Ver. 2006-09-21 ************

 Run condition is as follows
   Input  K5 file           : F:¥0202151500a.dat
   K5 file after edit       : F:¥0202151500a.dat.edit
   Sampling Frequency (MHz) : 4.000000
   AD bits                  : 1
   # of channels            : 4
   Start Time               : 15:15:01 (54901 seconds)
   Search header pattern    : FFFF???? 8B????FF

 Is this condition OK? ([Y]/N)     ＜＝ＯＫならリターンキー
```

リターンキーを押すと、実際のヘッダー位置のサーチが以下のように始まります。１０個までのサーチ結果（バイト位置と前のヘッダー位置からの差）を表示します。ヘッダー間隔が与えられたサンプリングモードで計算されるバイト数（＋ヘッダーバイト）の場合、一番右側にGOODと表示されます。

```
FILE : F:¥0202151500a.dat  (598007808 bytes)
Pattern FFFF0000 8B0000FF
Mask    FFFF0000 FF0000FF searching....
    0 Pattern found FFFFFF9B 8B1AD7FF at pos=      865584  sa=    865584
    1 Pattern found FFFFFFB3 8B1AD7FF at pos=     2865592  sa=   2000008 GOOD
    2 Pattern found FFFFFF3D 8B1AD7FF at pos=     4865600  sa=   2000008 GOOD
    3 Pattern found FFFFFFCD 8B1AD7FF at pos=     6865608  sa=   2000008 GOOD
    4 Pattern found FFFFFF03 8B1AD7FF at pos=     8865616  sa=   2000008 GOOD
```

```
    5 Pattern found FFFFFFDF 8B1AD7FF at pos=    10865624  sa=   2000008 GOOD
    6 Pattern found FFFFFF79 8B1AD7FF at pos=    12865632  sa=   2000008 GOOD
    7 Pattern found FFFFFF65 8B1AD7FF at pos=    14865640  sa=   2000008 GOOD
    8 Pattern found FFFFFF12 8B1AD7FF at pos=    16865648  sa=   2000008 GOOD
    9 Pattern found FFFFFF87 8B1AD7FF at pos=    18865656  sa=   2000008 GOOD
 Found byte offset is 865584

 Continue Processing with offset 865584? ([Y]/N)  ＜＝処理を続けるならリターン
```

この場合は、ファイルの最初がヘッダーデータではなかった場合です。リターンキーを押すと、

```
  1 -- Header Check Only using 865584 byte offset
  2 -- Replace Header    using 865584 byte offset

 Enter your selection ([1]/2)
```

ここでヘッダーチェックだけをやりたい場合はそのままリターンです。

すると、ヘッダーに相当する部分のデータ（６４ビット）が以下のように３２ビット整数×２の１６進表示で表示されます。

```
・・・・・・・・・・・・・・
0027     FFFFFE4C 8B1AD7FF
0028     FFFFFF1F 8B1AD7FF
0029     FFFFFE2D 8B1AD7FF
0030     FFFFFEFF 8B1AD7FF
0031     FFFFFEA0 8B1AD7FF
0032     FFFFFFFD 8B1AD7FF
0033     FFFFFF2E 8B1AD7FF
0034     FFFFFF6E 8B1AD7FF
・・・・・・・・・・・・・・
```

上記のように同じようなパターンが繰り返され表示されていれば、正常にヘッダー位置を認識できていると言えます。プログラムはすべてのヘッダー位置データを表示してストップします。

２を選択すると、以下の例のように実際の変換がスタートします。(修復前のヘッダーと修復後のヘッダーがそれぞれ１６進数表示されます。)

```
 Enter your selection ([1]/2) 2
FILE : F:¥0202151500a.dat  (598007808 bytes)

Frm#       OLD HEADER              NEW HEADER     (HH:MM:SS)
----   -----------------       -----------------  ----------
0000   FFFFFF9B 8B1AD7FF ===>  FFFFFFFF 8B1AD675  (15:15:01)  Header corrected
```

```
0001   FFFFFFB3 8B1AD7FF ===>  FFFFFFFF 8B1AD676  (15:15:02)  Header corrected
0002   FFFFFF3D 8B1AD7FF ===>  FFFFFFFF 8B1AD677  (15:15:03)  Header corrected
0003   FFFFFFCD 8B1AD7FF ===>  FFFFFFFF 8B1AD678  (15:15:04)  Header corrected
0004   FFFFFF03 8B1AD7FF ===>  FFFFFFFF 8B1AD679  (15:15:05)  Header corrected
0005   FFFFFFDF 8B1AD7FF ===>  FFFFFFFF 8B1AD67A  (15:15:06)  Header corrected
0006   FFFFFF79 8B1AD7FF ===>  FFFFFFFF 8B1AD67B  (15:15:07)  Header corrected
    ・・・・・・・・・
0294   FFFFFF27 8B1AD7FF ===>  FFFFFFFF 8B1AD79B  (15:19:55)  Header corrected
0295   FFFFFF70 8B1AD7FF ===>  FFFFFFFF 8B1AD79C  (15:19:56)  Header corrected
0296   FFFFFE5D 8B1AD7FF ===>  FFFFFFFF 8B1AD79D  (15:19:57)  Header corrected
0297   FFFFFF75 8B1AD7FF ===>  FFFFFFFF 8B1AD79E  (15:19:58)  Header corrected
0298   FFFFFFD8 8B1AD7FF ===>  FFFFFFFF 8B1AD79F  (15:19:59)  Header corrected

  Total Frame = 299   Corrected Frame = 299
  Created File = F:¥0202151500a.dat.edit

Time elapsed for processing is 65.438000 sec
```

ヘッダーが修復されたデータファイルはこの例の場合は F:¥0202151500a.dat.edit となります。

実行例　（機能２）

```
data_recov rdv58_ts2_242-1800a_200642180000.k5

FILE : rdv58_ts2_242-1800a_2006242180000.k5  (2632056832 bytes)


************ data_recov  Ver. 2006-09-21 ************

 Run condition is as follows
   Input  K5 file           : rdv58_ts2_242-1800a_2006242180000.k5
   K5 file after edit       : rdv58_ts2_242-1800a_2006242180000.k5.edit
   Sampling Frequency (MHz) : 16.000000
   AD bits                  : 1
   # of channels            : 4
   Start Time               : 17:59:31 (64771 seconds)
   Offset Bytes             : 0

 Is this condition OK? ([Y]/N)   ＜＝＝　ここはリターンのみ

  1 -- Header Check Only using 0 byte offset
  2 -- Replace Header    using 0 byte offset

 Enter your selection ([1]/2) 2 ＜＝＝　ここで２を入力

FILE : rdv58_ts2_242-1800a_2006242180000.k5  (2632056832 bytes)
```

```
Frm#       OLD HEADER              NEW HEADER     (HH:MM:SS)
----   -----------------       -----------------  ----------
0000   FFFFFFFF 8B22FD03 ===>  FFFFFFFF 8B22FD03  (17:59:31)  OK
0001   FFFFFFFF 8B22FD04 ===>  FFFFFFFF 8B22FD04  (17:59:32)  OK
0002   FFFFFFFF 8B22FD05 ===>  FFFFFFFF 8B22FD05  (17:59:33)  OK
0003   FFFFFFFF 8B22FD06 ===>  FFFFFFFF 8B22FD06  (17:59:34)  OK
0004   FFFFFFFF 8B22FD07 ===>  FFFFFFFF 8B22FD07  (17:59:35)  OK
0005   FFFFFFFF 8B22FD08 ===>  FFFFFFFF 8B22FD08  (17:59:36)  OK
  . . . . . .
0065   FFFFFFFF 8B22FD44 ===>  FFFFFFFF 8B22FD44  (18:00:36)  OK
0066   FFFFFFFF 8B22FD45 ===>  FFFFFFFF 8B22FD45  (18:00:37)  OK
0067   FFFFFFFF 8B22FD46 ===>  FFFFFFFF 8B22FD46  (18:00:38)  OK
0068   8B22FD47 AA6E16AB ===>  FFFFFFFF 8B22FD47  (18:00:39)  Header corrected
0069   FFFFFFFF 8B22FD48 ===>  FFFFFFFF 8B22FD48  (18:00:40)  OK
0070   FFFFFFFF 8B22FD49 ===>  FFFFFFFF 8B22FD49  (18:00:41)  OK
 . . . . . .
0328   FFFFFFFF 8B22FE4B ===>  FFFFFFFF 8B22FE4B  (18:04:59)  OK
0329   FFFFFFFF 8B22FE4C ===>  FFFFFFFF 8B22FE4C  (18:05:00)  OK

  Total Frame = 330   Corrected Frame = 1
  Created File = rdv58_ts2_242-1800a_2006242180000.k5.edit

Time elapsed for processing is 42.390000 sec
```