

K5 データ修復ソフト (data_recov) 説明書

Ver. 2007-10-23

T. Kondo

今回の変更点：ヘッダー修復プログラム data_recov を VSSP32 にも対応させ、さらに非会話型の実行を可能にしたことを反映させるためのマニュアルの改修。

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 および K5/VSSP32 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 および K5/VSSP32 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ための調査）。

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： FFFFFFFx 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。

※現在は VSSP に対するサーチパターンは“FFFF???? 8B????FF”

VSSP32 に対しては“FFFF???? 8C??????”としている

使用上の注意：

1. サンプルングパラメータ、データ開始時刻はユーザーが与える必要があります。(最初のデータが正常の場合は必要ありません)
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 [stype] K5file [sfreq adbit numch start_time [HDpattern1 HDpattern2]]  
[option]
```

ここで stype ----- サンプラータイプ '32' :VSSP32。省略時は VSSP

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 ----- A/D 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)

option (順不同可) は以下のとおり

-recov[er] ---- 非会話型で修復まで行う
-check ----- 非会話型でチェックのみ行う
-pat1 HDpattern1 --- 異常ヘッダーサーチパターン 1 のセット
-pat2 HDpattern2 --- 異常ヘッダーサーチパターン 2 のセット
-vssp32 ----- サンプラータイプを VSSP32 にセットする
(デフォルトは VSSP)
-out ofile --- 出力ファイル名をセットする
-in k5file --- 修復する K5 ファイル名を指定する
-offset nbytes -- ヘッダー位置のオフセットをバイト単位で強制的
にセットする

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オプションで指定しない限り
K5 ファイル名 + ".edit" となります。

実行例 1 (機能 1)

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

```
***** data_recov Ver. 2007-10-20 *****
```

Run condition is as follows

```
Sampler Type          : VSSP  
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 FFFFFFF9B 8B1AD7FF at pos=	865584	sa=	865584
1	Pattern found FFFFFFFB3 8B1AD7FF at pos=	2865592	sa=	2000008 GOOD
2	Pattern found FFFFFFF3D 8B1AD7FF at pos=	4865600	sa=	2000008 GOOD
3	Pattern found FFFFFFFCD 8B1AD7FF at pos=	6865608	sa=	2000008 GOOD
4	Pattern found FFFFFFF03 8B1AD7FF at pos=	8865616	sa=	2000008 GOOD
5	Pattern found FFFFFFFDF 8B1AD7FF at pos=	10865624	sa=	2000008 GOOD
6	Pattern found FFFFFFF79 8B1AD7FF at pos=	12865632	sa=	2000008 GOOD
7	Pattern found FFFFFFF65 8B1AD7FF at pos=	14865640	sa=	2000008 GOOD
8	Pattern found FFFFFFF12 8B1AD7FF at pos=	16865648	sa=	2000008 GOOD
9	Pattern found FFFFFFF87 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 FFFFFFFE4C 8B1AD7FF
0028 FFFFFFF1F 8B1AD7FF

```

0029  FFFFFFFE2D 8B1AD7FF
0030  FFFFFFFEFF 8B1AD7FF
0031  FFFFFFFEA0 8B1AD7FF
0032  FFFFFFFFD 8B1AD7FF
0033  FFFFFFF2E 8B1AD7FF
0034  FFFFFFF6E 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)

```

Fr#	OLD HEADER	NEW HEADER	(HH:MM:SS)	
0000	FFFFFF9B 8B1AD7FF ==>	FFFFFFF 8B1AD675	(15:15:01)	Header corrected
0001	FFFFFFB3 8B1AD7FF ==>	FFFFFFF 8B1AD676	(15:15:02)	Header corrected
0002	FFFFFF3D 8B1AD7FF ==>	FFFFFFF 8B1AD677	(15:15:03)	Header corrected
0003	FFFFFFCD 8B1AD7FF ==>	FFFFFFF 8B1AD678	(15:15:04)	Header corrected
0004	FFFFFF03 8B1AD7FF ==>	FFFFFFF 8B1AD679	(15:15:05)	Header corrected
0005	FFFFFFDF 8B1AD7FF ==>	FFFFFFF 8B1AD67A	(15:15:06)	Header corrected
0006	FFFFFF79 8B1AD7FF ==>	FFFFFFF 8B1AD67B	(15:15:07)	Header corrected
.				
0294	FFFFFF27 8B1AD7FF ==>	FFFFFFF 8B1AD79B	(15:19:55)	Header corrected
0295	FFFFFF70 8B1AD7FF ==>	FFFFFFF 8B1AD79C	(15:19:56)	Header corrected
0296	FFFFFFE5D 8B1AD7FF ==>	FFFFFFF 8B1AD79D	(15:19:57)	Header corrected
0297	FFFFFF75 8B1AD7FF ==>	FFFFFFF 8B1AD79E	(15:19:58)	Header corrected
0298	FFFFFFD8 8B1AD7FF ==>	FFFFFFF 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 2 (機能 2)

```
data_recov rdv58_ts2_242-1800a_200642180000.k5
```

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

***** data_recov Ver. 2007-10-20 *****

Run condition is as follows

Sampler Type : VSSP
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

実行例 3 K5/VSSP32 データを非会話型で修復

```
data_recov 32 errorGSlsample.k5 -recover
または
data_recov errorGSlsample.k5 -recover -vssp32
```

以下自動的に最後まで実行される

```
***** data_recov Ver. 2007-10-20 *****
```

Run condition is as follows

```
Sampler Type      : VSSP32
Input K5 file     : errorGSlsample.k5
K5 file after edit : errorGSlsample.k5.edit
Sampling Frequency (MHz) : 0.000000
AD bits           : 0
# of channels     : 0
Start Time        : 00:00:00 (0 seconds)
Search header pattern : FFFF???? 8C??????
```

FILE : errorGSlsample.k5 (616038400 bytes)

Pattern FFFF0000 8C000000

Mask FFFF0000 FF000000 searching....

0	Pattern found FFFFFFFF 8C22EE4D at pos=	0	sa=	0
1	Pattern found FFFFFFFF 8C22EE4E at pos=	8000032	sa=	8000032
2	Pattern found FFFFFFFF 8C22EE4F at pos=	16000064	sa=	8000032
3	Pattern found FFFFFFFF 8C22EE50 at pos=	24000096	sa=	8000032
4	Pattern found FFFF8769 8C04A773 at pos=	27583560	sa=	3583464
5	Pattern found FFFFFFFF 8C22EE52 at pos=	40000160	sa=	12416600
6	Pattern found FFFFFFFF 8C22EE53 at pos=	48000192	sa=	8000032
7	Pattern found FFFFFFFF 8C22EE55 at pos=	64000256	sa=	16000064
8	Pattern found FFFF80A8 8C5FA971 at pos=	64502988	sa=	502732
9	Pattern found FFFFFFFF 8C22EE56 at pos=	72000288	sa=	7497300

Found byte offset is 0

FILE : errorGSlsample.k5 (616038400 bytes)

This is K5/VSSP32 format data.

1st Extended Header Info is as follows.

Year = 2007 Total Day = 290

Version Major = 1 Minor = 0 # of AUX Bytes = 20

AUX Field Data : 20540801 4B555354 32334255 7374356B 20202034

Format # = 1

Filter (MHz) : 8

Station info : ID2(T) name(TSUKUB32)

PC host name : k5ts4

FILE : errorGSlsample.k5 (616038400 bytes)

Frm#	OLD HEADER	NEW HEADER	(HH:MM:SS)	
0000	FFFFFFFF 8C22EE4D ==>	FFFFFFFF 8C22EE4D	(16:56:45)	OK
0001	FFFFFFFF 8C22EE4E ==>	FFFFFFFF 8C22EE4E	(16:56:46)	OK
0002	FFFFFFFF 8C22EE4F ==>	FFFFFFFF 8C22EE4F	(16:56:47)	OK
0003	FFFFFFFF 8C22EE50 ==>	FFFFFFFF 8C22EE50	(16:56:48)	OK
0004	FFFFFFFF 0F22EE51 ==>	FFFFFFFF 8C22EE51	(16:56:49)	Header corrected
0005	FFFFFFFF 8C22EE52 ==>	FFFFFFFF 8C22EE52	(16:56:50)	OK
0006	FFFFFFFF 8C22EE53 ==>	FFFFFFFF 8C22EE53	(16:56:51)	OK
0007	EE76FFFF 0F22AC02 ==>	FFFFFFFF 8C22EE54	(16:56:52)	Header corrected
0008	FFFFFFFF 8C22EE55 ==>	FFFFFFFF 8C22EE55	(16:56:53)	OK
0009	FFFFFFFF 8C22EE56 ==>	FFFFFFFF 8C22EE56	(16:56:54)	OK
0010	FFFFFFFF 8C22EE57 ==>	FFFFFFFF 8C22EE57	(16:56:55)	OK
0011	EE58FFFF 0F228C22 ==>	FFFFFFFF 8C22EE58	(16:56:56)	Header corrected
0012	FFFFFFFF 8C22EE59 ==>	FFFFFFFF 8C22EE59	(16:56:57)	OK
0013	FFFFFFFF 8C22EE5A ==>	FFFFFFFF 8C22EE5A	(16:56:58)	OK
			
0053	FFFFFFFF 8C22EE82 ==>	FFFFFFFF 8C22EE82	(16:57:38)	OK
0054	FFFFFFFF 8C22EE83 ==>	FFFFFFFF 8C22EE83	(16:57:39)	OK
0055	FFFFFFFF 8C22EE84 ==>	FFFFFFFF 8C22EE84	(16:57:40)	Header corrected
0056	FFFFFFFF 8C22EE85 ==>	FFFFFFFF 8C22EE85	(16:57:41)	OK
0057	FFFFFFFF 8C22EE86 ==>	FFFFFFFF 8C22EE86	(16:57:42)	OK
0058	FFFFFFFF 8C22EE87 ==>	FFFFFFFF 8C22EE87	(16:57:43)	Header corrected
0059	FFFFFFFF 8C22EE88 ==>	FFFFFFFF 8C22EE88	(16:57:44)	OK
0060	FFFFFFFF 8C22EE89 ==>	FFFFFFFF 8C22EE89	(16:57:45)	OK
0061	FFFFFFFF 8C22EE8A ==>	FFFFFFFF 8C22EE8A	(16:57:46)	OK
0062	FFFFFFFF 8C22EE8B ==>	FFFFFFFF 8C22EE8B	(16:57:47)	OK
0063	FFFFFFFF 8C22EE8C ==>	FFFFFFFF 8C22EE8C	(16:57:48)	OK
0064	FFFFFFFF 8C22EE8D ==>	FFFFFFFF 8C22EE8D	(16:57:49)	OK
0065	FFFFFFFF 8C22EE8E ==>	FFFFFFFF 8C22EE8E	(16:57:50)	Header corrected
0066	FFFFFFFF 8C22EE8F ==>	FFFFFFFF 8C22EE8F	(16:57:51)	OK
0067	FFFFFFFF 8C22EE90 ==>	FFFFFFFF 8C22EE90	(16:57:52)	OK
0068	FFFFFFFF 8C22EE91 ==>	FFFFFFFF 8C22EE91	(16:57:53)	OK
0069	FFFFFFFF 8C22EE92 ==>	FFFFFFFF 8C22EE92	(16:57:54)	OK
0070	FFFFFFFF 8C22EE93 ==>	FFFFFFFF 8C22EE93	(16:57:55)	OK
0071	FFFFFFFF 8C22EE94 ==>	FFFFFFFF 8C22EE94	(16:57:56)	OK
0072	FFFFFFFF 8C22EE95 ==>	FFFFFFFF 8C22EE95	(16:57:57)	OK
0073	FFFFFFFF 8C22EE96 ==>	FFFFFFFF 8C22EE96	(16:57:58)	OK
0074	FFFFFFFF 8C22EE97 ==>	FFFFFFFF 8C22EE97	(16:57:59)	Header corrected
0075	FFFFFFFF 8C22EE98 ==>	FFFFFFFF 8C22EE98	(16:58:00)	OK
0076	FFFFFFFF 8C22EE99 ==>	FFFFFFFF 8C22EE99	(16:58:01)	OK
0077	FFFFFFFF 8C22EE9A ==>	FFFFFFFF 8C22EE9A	(16:58:02)	OK

Total Frame = 78 Corrected Frame = 16
Created File = errorGSlsample.k5.edit

Time elapsed for processing is 111.580000 sec