

# K5/VSSP (IP-VLBI) 相関処理手引き

T.KONDO

2019 年 12 月 21 日改訂

*T. Kondo*

## 目 次

<b>1</b>	<b>ソフトウェアのインストール法</b>	<b>4</b>
1.1	アーカイブファイルからの一括インストール	4
1.2	個別インストール	4
1.3	インストール後のディレクトリ・ファイル構成	5
1.4	環境の設定	6
<b>2</b>	<b>ソフトウェア一覧</b>	<b>7</b>
2.1	予測値計算関係	7
2.2	相関処理関係	7
2.3	データチェック関係	7
2.4	データフォーマット変換関係	8
<b>3</b>	<b>予測値計算関係ソフトの使い方</b>	<b>9</b>
3.1	apri.calc	9
3.1.1	実行方法	9
3.1.2	実行例 (環境変数を確認・スケジュールファイルの中身確認)	14
3.1.3	実行例 (K5/VSSP データ処理の場合)	17
3.1.4	実行例 (VDIF フォーマットデータ同志の相関処理の場合)	22
3.1.5	実行例 (Mark-5B フォーマットデータ同志の相関処理の場合)	23
3.1.6	実行例 (X 局データが VSSP32、Y 局データが VDIF フォーマットデータの場合)	24
3.1.7	実行例 (X 局、Y 局ともに VDIF データ、ただし Y 局データは単 CH/スレッドのマルチスレッドデータの場合)	25
3.2	skdchk	27
3.2.1	実行方法	28
3.2.2	実行例	28
<b>4</b>	<b>相関処理関係ソフトの使い方</b>	<b>30</b>
4.1	fx_cor, fx_cor_new, cor, cor_new	30
4.1.1	実行方法	30
4.1.2	実行例: cor (1 ビットサンプリングデータ専用) による相関処理	37
4.1.3	実行例: fx_cor_new による相関処理 (Mark5B データ同志の処理)	40
4.2	fx_cor_all, fx_cor_all_new, cor_all, cor_all_new	42

4.2.1	実行方法 . . . . .	42
4.3	sdelay . . . . .	45
4.3.1	実行方法 . . . . .	45
4.3.2	実行例 . . . . .	47
4.3.3	出力ファイルフォーマット . . . . .	53
4.4	cor_mon . . . . .	54
4.4.1	実行方法 . . . . .	54
4.4.2	実行例 . . . . .	55
<b>5</b>	<b>データチェック関係ソフトの使い方</b>	<b>56</b>
5.1	oscillo . . . . .	56
5.1.1	実行方法 . . . . .	56
5.1.2	実行例 . . . . .	56
5.2	speana . . . . .	57
5.2.1	実行方法 . . . . .	57
5.2.2	実行例 . . . . .	58
5.3	g_speana . . . . .	58
5.3.1	実行方法 . . . . .	58
5.3.2	実行例 . . . . .	60
5.4	datachk . . . . .	61
5.4.1	実行方法 . . . . .	61
5.4.2	サマリーファイル . . . . .	62
5.4.3	エラーログファイル . . . . .	63
5.5	vdifcheck . . . . .	63
5.5.1	実行方法 . . . . .	63
5.5.2	実行例 . . . . .	64
5.6	m5check . . . . .	65
5.6.1	実行方法 . . . . .	65
5.6.2	実行例 . . . . .	65
<b>6</b>	<b>データフォーマット変換ソフトの使い方</b>	<b>67</b>
6.1	k5tom5b . . . . .	67
6.1.1	実行方法 . . . . .	67
6.1.2	実行例 . . . . .	68
6.2	k5tom5 . . . . .	69
6.2.1	How to execute . . . . .	69
6.2.2	実行例 . . . . .	70
6.3	k5tovdif . . . . .	74
6.3.1	実行方法 . . . . .	74
6.4	ads2k5 . . . . .	75
6.4.1	実行方法 . . . . .	76
6.5	m5btok5 . . . . .	77
6.5.1	実行方法 . . . . .	77
6.6	m5tok5 . . . . .	79
6.6.1	実行方法 . . . . .	79
6.7	vdif2k5 . . . . .	81

6.7.1	実行方法 . . . . .	81
6.8	vdif2m5b . . . . .	82
6.8.1	実行方法 . . . . .	82
7	相関処理の実際 . . . . .	84
7.1	フリンジサーチ . . . . .	84
7.2	全観測 (スキャン) の処理 . . . . .	88
7.3	K5/VSSP 以外のフォーマットデータ (VDIF, Mark5B, ADS) を含む場合の相関処理 . . . . .	89
8	文書更新履歴 . . . . .	90
A	予測値ファイルフォーマット . . . . .	91
B	K5 ソフトウェア相関器出力フォーマット (FORMAT 7 形式) . . . . .	101
C	KSP 型相関データファイルフォーマット . . . . .	104
C.1	KSP 相関データフォーマットの拡張 . . . . .	104
C.2	相関生データファイルデータフォーマット: ヘッダーレコード (HD) . . . . .	105
C.3	相関生データファイルデータフォーマット: ユニット毎の相関積分データ (UD) . . . . .	107
C.4	相関生データファイルデータフォーマット (CRSMODE="F" の場合のみ): ユニット毎の相関積分データ (UD) . . . . .	109
D	各種 VLBI 観測データフォーマット . . . . .	112
D.1	データフォーマットの種類 . . . . .	112
D.2	K5/VSSP, VSSP32 および VSSP64 フォーマット . . . . .	113
D.2.1	データ構造 . . . . .	113
D.2.2	VSSP ヘッダーフォーマット仕様 . . . . .	113
D.2.3	VSSP32 および VSSP64 ヘッダーフォーマット一般仕様 . . . . .	114
D.2.4	VSSP32 ヘッダーフォーマット番号別仕様 . . . . .	115
D.2.5	サンプラー出力データフォーマット . . . . .	119
D.2.6	拡張データフォーマット その 1 . . . . .	120
D.2.7	拡張データフォーマット その 2 . . . . .	121
D.3	Mark5B データフォーマット . . . . .	121
D.3.1	データ構造 . . . . .	121
D.3.2	Mark5B ヘッダー部 . . . . .	121
D.3.3	Mark5B データ部 . . . . .	121
D.4	VDIF フォーマット . . . . .	123
D.4.1	データ構造 . . . . .	123
D.4.2	VDIF ヘッダー部 . . . . .	123
D.4.3	VDIF データ部 . . . . .	125
E	PGPLOT のインストール法 . . . . .	126
E.1	FreeBSD の port を使用する方法 . . . . .	126
E.2	ステップバイステップ法 . . . . .	126
F	FFTW のインストール法 . . . . .	128

## 1 ソフトウェアのインストール法

グラフィック表示に PGPLOT または GNUPLOT を使用するので、予め PGPLOT または GNUPLOT がインストールされている必要がある。また FFT 計算に FFTW パッケージを利用する場合はあらかじめ FFTW パッケージがインストールされている必要がある。PGPLOT および FFTW のインストール法は付録の E 章および F 章を参考にされたし。

### 1.1 アーカイブファイルからの一括インストール

適当なディレクトリ (\$HOME/K5 とする) に ipvlbi\_corYYYYMMDD.tar.gz (ここで YYYYMMDD はアーカイブ年月日情報) を置き

```
tar xvzf ipvlbi_corYYYYMMDD.tar.gz
```

で解凍する。ipvlbiYYYYMMDD ディレクトリが出来るので

```
cd ipvlbiYYYYMMDD
```

で作業ディレクトリを移動し、そのディレクトリで make を実行する。具体的な make の実行法は以下の通り

```
make T=COR [F=FFTW] [G=GNUPLOT]
```

ここで F=FFTW --- 相関処理ソフトやスペクトル表示ソフトで FFT に FFTW パッケージを使用  
G=GNUPLOT --- グラフィック表示に GNUPLOT を使用する場合 (デフォルトは PGPLOT)

FFTW パッケージがすでにインストールされている場合は F=FFTW オプションを使う事で高速な FFT が使用できるようになる。

コンパイルし直したい場合は

```
make clean
```

の後、

```
make T=COR F=FFTW
```

のように実行する。

### 1.2 個別インストール

アーカイブを展開したディレクトリを \$HOME/K5 とすると

```
cd $HOME/K5/ipvlbiYYYYMMDD/apri
make
make install
cd $HOME/K5/ipvlbiYYYYMMDD/corr
make F=FFTW <== FFTW パッケージを使用するとき
make install
cd $HOME/K5/ipvlbiYYYYMMDD/sdelay
make
make install
cd $HOME/K5/ipvlbiYYYYMMDD/src
make S=NONE install <== サンプラーを使用しない設定
cd $HOME/K5/ipvlbiYYYYMMDD/mark5
make
make install
cd $HOME/K5/ipvlbiYYYYMMDD/vdif
make
make install
```

src ディレクトリの make のし方が他のディレクトリとは異なる (install も同時に指定する) ので注意のこと! それぞれのディレクトリで make し直したいときは、それぞれのディレクトリで

```
make clean
```

の後、 make を実行する。

## 1.3 インストール後のディレクトリ・ファイル構成

```

$HOME/K5/ipv1biYYYYMMDD/
|
+readme.txt          説明書
+archive_cor         アーカイブ作成用シェルスクリプト
+pgplot_install.txt  PGPlot をインストールする際の説明書
+fftw_install.txt    FFTW をインストールする際の説明書
+makefile            K5 ユーティティソフトインストール用 makefile
|
+apri/              ... 予測値計算ソフト用ディレクトリ
|   +*.c            ユーティリティプログラム (C ソース)
|   +*.h            インクルードファイル
|   +makefile       ユーティリティプログラム用 makefile
|
+bin/              ... 実行形式ファイル用ディレクトリ
|
+corr/            ... 相関処理ソフト用ディレクトリ
|   +*.c          ユーティリティプログラム (C ソース)
|   +*.h          インクルードファイル
|   +makefile     ユーティリティプログラム用 makefile
|
+corrapri/        ... 予測値ファイル出力用ディレクトリ
|   +ape_sample.txt   サンプル予測値ファイル
|   +apelist_sample.txt サンプル予測値ファイルリスト
|
+cout/            ... 相関処理結果出力用ディレクトリ
|   +coutNNNN.txt    XF 処理出力ファイル
|   +couttNNNN.txt   FX 処理出力ファイル
|
+log/             ... ログ出力用ディレクトリ
|
+mark5/          ... Mark5 データ変換ソフト用ディレクトリ
|   +*.c          ユーティリティプログラム (C ソース)
|   +*.h          インクルードファイル
|   +makefile     ユーティリティプログラム用 makefile
|   +m5tok5info.txt  Mrak5 から K5 への変換情報サンプルファイル
|   +k5tom5info.txt  K5 から Mrak5 への変換情報サンプルファイル
|   +vlbainfo.txt    VLBA から Mrak5 への変換情報サンプルファイル
|
+sdelay/         ... 粗決定サーチソフト用ディレクトリ
|   +*.c          ユーティリティプログラム (C ソース)
|   +*.h          インクルードファイル
|   +makefile     ユーティリティプログラム用 makefile
|
+sked/           ... スケジュールファイル用ディレクトリ
|   +sample.skd    サンプルスケジュールファイル
|   +sample.vex    サンプル VEX ファイル
|
+src/            ... 観測ソフト用ディレクトリ
|   +*.c          ユーティリティプログラム (C ソース)
|   +*.h          インクルードファイル
|   +makefile     ユーティリティプログラム用 makefile
|
+vdif/           ... VDIFF データ変換ソフト用ディレクトリ
|   +*.c          ユーティリティプログラム (C ソース)
|   +*.h          インクルードファイル
|   +makefile     ユーティリティプログラム用 makefile
|
+man/            ... man ページ用ディレクトリ
|
+doc/            ... ドキュメント用ディレクトリ
|   +cor_manual_j.YYYYYMMDD.pdf  日本語マニュアル
|   +cor_manual.YYYYYMMDD.pdf   英語マニュアル
|   +index.html                 日本語 Web マニュアル
|   +index-e.html               英語 Web マニュアル

```

## 1.4 環境の設定

1. インストールしたソフト群のディレクトリが例えば\$HOME/K5/ipv1biXXXXXXXXX である場合 \$HOME/ipv1bi にシンボリックリンクを貼る。

```
ln -s $HOME/K5/ipv1biXXXXXXXXX $HOME/ipv1bi
```

すでにリンクがある場合新しく貼り直す場合は “f” オプションをつけて、

```
ln -sf $HOME/K5/ipv1biXXXXXXXXX $HOME/ipv1bi
```

2. 環境変数 PATH に\$HOME/ipv1bi/bin を追加する。

```
export PATH=$PATH:$HOME/ipv1bi/bin      (.bashrc の場合)
```

```
setenv PATH $PATH:$HOME/ipv1bi/bin      (.cshrc の場合)
```

3. 環境変数 MANPATH に\$HOME/ipv1bi/man を追加する。

```
export MANPATH=$HOME/ipv1bi/man:$MANPATH      (.bashrc の場合)
```

```
setenv MANPATH $HOME/ipv1bi/man:$MANPATH      (.cshrc の場合)
```

Ver.2010-02-08 以降のアーカイブでは man ページファイルも含まれているので、この設定を行なっていれば、man コマンドで各ソフトの使用方法を得ることができる。また “man k5cor” を実行することにより、K5 関連処理ソフト群全体の説明を得ることができる。

以上の設定を行なっておけば、K5 ソフト群のバージョンアップを行った際も 1. でシンボリックリンクを貼り直すだけで良くなる。

## 2 ソフトウェア一覧

ここに示すソフトウェアは“sdelay”を除きプログラム名 [リターン] で使用方法が表示される。またプログラム名 --version [リターン] でバージョンが表示される。

### 2.1 予測値計算関係

apri_calc	– 予測値ファイル作成。スケジュールファイル (SKED 形式または VEX 形式) を読み、相関処理に必要な予測値を計算し予測値ファイルの作成を行なう。	...apri
skdchk	– スケジュールファイルチェック。スケジュールファイルを読み、局情報、電波星情報、観測情報を出力する。最後に、全観測時間を秒で示すと共に、必要なディスク容量を局毎に表示する。	...src

注：最後の項はソースファイルのあるディレクトリ

### 2.2 相関処理関係

fx_cor	– FX 方式での相関処理。予測値ファイル使用した相互相関処理、予測値ファイルを使用しない自己相関処理を実行する。	...corr
fx_cor_all	– FX 方式での相関処理。予測値ファイル名一覧を収納したファイルを使用して複数の予測値ファイルに対する相互相関処理を実行する。	...corr
fx_cor_new	– FX 方式での相関処理。各種サンプリングデータフォーマット (K5/VSSP, VDIF, Mark5B, ADS) をサポート。	...corr
fx_cor_all_new	– fx_cor_all の各種フォーマット対応版	...corr
cor	– XF 方式での相関処理 (1 ビットサンプリングデータ専用)。予測値ファイル使用した相互相関処理、予測値ファイルを使用しない自己相関処理を実行する。	...corr
cor_all	– XF 方式での相関処理 (1 ビットサンプリングデータ専用)。予測値ファイル名一覧を収納したファイルを使用して複数の予測値ファイルに対する相互相関処理を実行する。	...corr
cor_new	– XF 方式相関処理 (1 ビットサンプリングデータ専用)。各種サンプリングデータフォーマット (K5/VSSP, VDIF, Mark5B, ADS) をサポート。	...corr
cor_all_new	– cor_all の各種フォーマット対応版	...corr
sdelay	– 粗決定サーチ。相関処理データから粗決定サーチを行ない遅延残差および遅延変化率残差を決定する。	...sdelay
cor_mon	– 相関関数のダイナミック表示を行なう (PGPLOT 専用 *)	...corr
* : PGPLOT 用にコンパイル時のみ作成される		

注：最後の項はソースファイルのあるディレクトリ。

### 2.3 データチェック関係

oscillo	– サンプリングデータのダイナミック表示 (k5/VSSP 用, PGPLOT 専用 *)。時系列データを連続的に表示する。	...src
---------	--	--------

speana	– スペクトラム表示 (k5/VSSP 用)。サンプリングデータのスペクトルを計算し表示する。	...src
speana_n	– スペクトラムのダイナミック表示 (k5/VSSP 用, PGPLOT 専用*)。サンプリングデータのスペクトルを計算し連続的に表示する。	...src
g_speana	– スペクトラム表示 (各種サンプリングデータフォーマット対応)。各種フォーマットのサンプリングデータのスペクトルを計算し表示する。	...corr
datachk	– VSSP フォーマットデータファイルチェック	...src
vdifcheck	– VDIF フォーマットデータファイルチェック	...vdif
m5check	– Mark-5 フォーマットデータファイルチェック	...mark5
* : PGPLOT 用にコンパイル時のみ作成される		

注 : 最後の項はソースファイルのあるディレクトリ

## 2.4 データフォーマット変換関係

k5tom5b	– K5 フォーマットから Mark5B フォーマットへの変換	...mark5
k5tom5	– K5 フォーマットから Mark5 フォーマットへの変換	...mark5
k5tovdif	– K5 フォーマットから VDIF フォーマットへの変換	...vdif
ads2k5	– ADS3000 (DBBC モード) フォーマットから K5 フォーマットへの変換	...mark5
m5btok5	– Mark5B フォーマットから K5 フォーマットへの変換	...mark5
m5tok5	– Mark5 フォーマットから K5 フォーマットへの変換	...mark5
vdif2k5	– VDIF フォーマットから K5 フォーマットへの変換	...vdif
vdif2m5b	– VDIF フォーマットから Mark5B フォーマットへの変換	...vdif

注 : 最後の項はソースファイルのあるディレクトリ



### 3 予測値計算関係ソフトの使い方

#### 3.1 apri\_calc

ユーティリティ名

apri\_calc

機能

スケジュールファイル (SKED 形式または VEX 形式) を読み、相関処理に必要な予測値を計算し予測値ファイルの作成を行なう。

Ver.2019-12-21 以降では SKED ファイル中の \$CODES ブロックの解析方法を改良し、周波数と CH# の対応が局毎に異なっている場合でも適切な予測値ファイルが作成できるようになっている。なお旧バージョンとの互換性を保つためにオプション ‘-legacy’ を追加した。

##### 3.1.1 実行方法

【方法 1】

apri\_calc *skdfile* [*options*]

ここで *skdfile*

- スケジュールファイル名 (VEX 又は SKED タイプ)  
“-” をファイル名の頭につけるとスケジュールファイルの中身のモニターだけを行う。

Ver.2016-10-12 以降ではオプション “-monit” も使える

*options* (順不同: パラメータ省略時は必要に応じて後で入力が必要される)

-apedir *apriori\_file\_out\_directory*

- 予測値ファイルを作るディレクトリを指定する  
無指定の場合は環境変数 K5APRIOUT でセットしたディレクトリ。  
環境変数がセットされていない場合は “../corrapi” が出力先となる

-baseid *baseline\_id*

- 基線 ID (2 文字または 4 文字)  
このオプションはモニターモード時にも有効 (ただし Ver. 2016-10-12 以降)

-coffset *clock\_offset*

- Y 局クロックオフセット (s)。  
正の値は X 局に対して進みを意味する。デフォルトは 0.0

-crate *clock\_rate*

- Y 局クロックレート (s/s)。デフォルトは 0.0

-ceepoch *epoch*

- クロックパラメータのエポックを設定する  
*epoch* のフォーマットは以下の 3 種類の形式のいずれか  
YYYY/DDD-HH:MM:SS または YYYY/MM/DD-HH:MM:SS  
または YYYYDDDDHHMMSS  
エポックをセットしない場合はそれぞれの PRT がエポックとなる

-g *group*

- 周波数グループ (1-4) PC に対応 (デフォルトは 1)  
X 局 Y 局ともにデータフォーマットを K5/VSSP 以外に設定した時はデフォルトは 0 (全 CH 処理) となる

-ch *channel*

- 1ch 観測モード時の周波数 CH を指定する (デフォルトは 1)

-start *start\_obs*

- 開始観測 # (デフォルトは 1)

-stop *stop\_obs*

- 終了観測 # (デフォルトは最後の観測)

-xdir *xdir*

- X 局データディレクトリ (デフォルトは “./”)

-ydir *ydir*

- Y 局データディレクトリ (デフォルトは “./”)

-ut1 *ut1\_c*

- UT1-UTC (sec)

-wobbx *wobbx*

- Wobb X (arcsec)

- wobby *wobby*      -    Wobb Y (arcsec)
- type1 | -type2      -    K5 観測データファイル命名則  
                          1: Type I      2: Type II  
                          (注: このコマンドは旧バージョンとの互換性を保つ目的で残しているが、次の "-type" コマンドの使用を推奨)  
                          -type1 : sidDDDNNNN.dat (default)  
                          -type2 : sidDDDHMMSSG.dat
- type *naming\_type*      -    K5 観測データファイル命名則  
                          1 : Type 1 sidDDDNNNN.dat (SKED 使用時のデフォルト)  
                          -1 : Type -1 sidDDDNNNN.#ch.dat  
                          2 : Type 2 sidDDDHMMSSG.dat  
                          -2 : Type -2 sidDDDHMMSSG.#ch.dat  
                          3 : Type 3 expid\_sidG\_scanid\_YYYYDDDDHHMMSS.k5  
                          (e-VLBI 準拠命名則)  
                          4 : 未使用  
                          5 : Type 5 expid\_scan#.stcode.k5a(-d) (VEX 使用時のデフォルト)  
                          命名則の詳細は補足 1 を参照のこと
- subnet | -nosubnet      -    サブネットモード制御  
                          デフォルトはサブネットモード ON ("-subnet")
- skey *satellite\_key*      -    「のぞみ」モードの衛星コードセット  
                          1 個以上のコードを設定するときは" "でくる。  
                          例: -skey "NOZ HYB HEO"
- xcoff *x\_clock\_offset*      -    X 局の時計の UTC に対するオフセット (sec)  
                          正の符号は UTC に対して進んでいることを意味する  
                          デフォルトは 0
- lsb                      -    サイドバンドを強制的に LSB にセットする (ただし SKED ファイル使用時のみ有効)
- shift *time*              -    最初のスキンの開始時刻を *time* で与えられた時刻にシフトする  
                          *time* は YYYY/DDD-HH:MM:SS または YYYY/MM/DD-HH:MM:SS または  
                          YYYYDDDDHHMMSS 形式で与える
- format VDIF|MK5|M5B|OCTAD|ADS|[VSSP] [*sampling\_cond*]      -    両局のデータフォーマットと必要であればサンプリング情報を設定する (デフォルトは VSSP)
- formX VDIF|MK5|M5B|OCTAD|ADS|[VSSP] [*sampling\_cond*]      -    X 局のデータフォーマットと必要であればサンプリング情報を設定する (デフォルトは VSSP)
- formY VDIF|MK5|M5B|OCTAD|ADS|[VSSP] [*sampling\_cond*]      -    Y 局のデータフォーマットと必要であればサンプリング情報を設定する (デフォルトは VSSP)  
                          ここで *sampling\_cond* は MK5, M5B, OCTAD, ADS フォーマットが指定された場合のサンプリング情報で以下の形式で記述する  
                           $xM[\text{Hz}]nB[\text{IT}]mC[\text{H}]$  (順不同)  
                           $x$  - サンプリング周波数 (MHz)

$n$  - AD ビット数  
 $m$  - チャンネル数

==== 以下のオプションは Ver.2016-10-12 以降で有効 ====

- source *star\_name*      - スケジュールから抽出するスキャンを電波源名で設定する  
このオプションはモニターモード時にも有効
- m[onit]                - モニターモード（スケジュールファイルのモニター）にセットする

==== 以下のオプションは Ver.2019-05-23 以降で有効 ====

- thread *thread\_num*    - VDIF データ (X,Y 局) の読み込みスレッド # を設定する。デフォルトはデータ中の最初のスレッド #
- threadX *thread\_num*   - X 局 VDIF データの読み込みスレッド # を設定する。デフォルトはデータ中の最初のスレッド #
- threadY *thread\_num*   - Y 局 VDIF データの読み込みスレッド # を設定する。デフォルトはデータ中の最初のスレッド #

==== 以下のオプションは Ver.2019-12-03 以降で有効 ====

- thread MULTI|*th1[,th2[,th3,...]]*
  - VDIF データ (X,Y 局) のチャンネル毎の読み込みスレッド # を設定する  
MULTI: CH#1, #2, ... のデフォルトスレッド # (0 からスタート) を設定する  
*th1[,th2[,th3,...]]*: チャンネル番号順にスレッド # を設定する。1 つしか設定しない場合には全 CH が同じスレッド # となる。
- threadX MULTI|*th1[,th2[,th3,...]]*
  - VDIF データ (X 局) のチャンネル毎の読み込みスレッド # を設定する
- threadY MULTI|*th1[,th2[,th3,...]]*
  - VDIF データ (Y 局) のチャンネル毎の読み込みスレッド # を設定する

==== 以下のオプションは Ver.2019-12-21 以降で有効 ====

- uoca                    - SKED ファイルの \$CODE ブロックの周波数と CH# の対応解析を旧方式で行う (Ver.2019-12-21 以前との互換性のため)。Ver.2019-12-21 以降は各局の周波数情報を厳密に解析している。( "uoca" は Use Old Code Analysis の意 )
- legacy                - オプション '-uoca' と同じ

### 【方法2】

```
apri_calc skdfile [apedir [baseid coffset roffset frqgr nob1 nob2 xdir ydir ut1_c wobbb wobbb
[naming_type [subnet [xcoff [source]]]]]
```

ここで *skdfile*

- スケジュールファイル名  
ファイル名の先頭に'-'を付けると、スケジュールファイルの中身のモニターだけを行う

*apedir*

- アプリオリファイル出力ディレクトリ  
省略時は 環境変数\$K5APRIOUT それがないときは../corrapi/

以下のパラメータを省略時は会話型となる

*baseid*

- 基線 I D ( 2 文字または 4 文字 )

*coffset*

- Y 局クロックオフセット (s)

<i>roffset</i>	– Y局クロックレート (s/s)
<i>frqgr</i>	– 周波数グループ (1-4) P C に対応
<i>nobs1</i>	– 開始観測 #
<i>nobs2</i>	– 終了観測 #
	<i>nobs1=0, nobs2=0</i> とすると可能なすべての観測番号
<i>xdir</i>	– X局データディレクトリ
<i>ydir</i>	– Y局データディレクトリ
<i>ut1_c</i>	– UT1-UTC (sec)
<i>wobbx</i>	– WobbX (arcsec)
<i>wobby</i>	– WobbX (arcsec)
<i>naming_type</i>	– K5 観測データファイル命名則
	1: Type I (SKED 使用時のデフォルト)    2: Type II
	-1: Type -I            -2: Type -II
	3: Type III (e-VLBI 命名則準拠)
	4: 未使用
	5: Type 5 (VEX 使用時のデフォルト)
	命名則の詳細は補足 1 を参照のこと
<i>subnet</i>	– 0: サブネットモードオフ、 1: サブネットモードオン (デフォルト)
<i>xcoff</i>	– X局の時計の UTC に対するオフセット (sec)
	正の符号は UTC に対して進んでいることを意味する
	デフォルトは 0
==== 以下のオプションは Ver.2016-10-11 以降で有効 ====	
<i>source</i>	– スケジュールから抽出するスキャンを電波源名で設定する

#### 【環境変数のモニター】

apri\_calc で使用する環境変数 K5SKED (スケジュールファイルディレクトリ)、K5APRIDIR (予測値ファイルのディレクトリ)、K5NAMING (ファイル命名則)、K5SATKEY (「のぞみ」モード観測時の衛星キーワード) の設定値とプログラムデフォルト値を確認する。

```
apri_calc env
```

#### 補足 1: 観測データファイルの命名則

Type 1 または -1 (I または -I)

XDDDNNNN.[#ch.]dat

ここで

X	– 局 ID (1 文字)
DDD	– 一番目のスキャンの通日
NNNN	– 観測 (スキャン) 番号 (4 桁)
#ch	– チャンネル数 (1 4) (負のタイプ時)

Type 2 または -2 (II または -II)

sidDDDDHHMMSSG.[#ch.]dat

ここで

sid	– 局 ID (1 文字か 2 文字)
-----	---------------------

DDD	－	スキャン開始通日 ( 3 桁 )
HH	－	スキャン開始時 ( 2 桁 )
MM	－	スキャン開始分 ( 2 桁 )
SS	－	スキャン開始秒 ( 2 桁 )
G	－	周波数グループ ID (a b c d) または null
#ch	－	チャンネル数 (1 4) (負のタイプ時)

## Type 3 e-VLBI ファイル命名則準拠

expid\_sidG\_scanid\_YYYYDDDDHHMMSS.k5

ここで

expid	－	実験コード
sid	－	局 I D (小文字 2 文字) 大文字の場合は小文字に変換
G	－	ターミナル (PC)ID (1 2 3 4) または null。 freqg パラメータで指定
scanid	－	スキャン I D VEX ファイル使用時はスキャン I Dそのまま SKED 使用時は ddd-hhmm 同じ分の中に複数の観測がある場合は 2 つ目以降に時間順に最後に a,b,c,d,... をつけていく
YYYYDDDDHHMMSS	－	観測開始時刻
.k5	－	K5 データ識別子

## Type 4 未使用

## Type 5 VEX ファイル使用時のデフォルト (旧バージョンとの互換性保持目的)

expid\_scan#.stcode.k5a(-d)

ここで

expid	－	実験コード
scan#	－	スキャン番号
stcode	－	局 ID
k5	－	固定値
a-d	－	周波数グループ (または null)

## 補足 2 : 作成されるアプリオリファイルの命名規則

## Type I

apeDDDNNNNXYG.txt

ここで	‘ape’	－	固定文字
	DDD	－	通日（ 2 日またがる場合も最初の通日）
	NNNN	－	観測番号（ 4 桁）
	XY	－	基線 I D（ 2 文字または 4 文字） （スケジュールファイル中の I D）
	G	－	P C の対応を示す 1 文字 (a,b,c,d) または null 周波数グループ 1-4 または 0 に対応

## Type II

apeDDDHHMMSSXYG.txt

ここで ‘ape’ – 固定文字  
 DDD – 通日 ( スキャン開始時 )  
 HH – 時 ( “ ” )  
 MM – 分 ( “ ” )  
 SS – 秒 ( “ ” )  
 XY – 基線 I D ( 2 文字または 4 文字 )  
       ( スケジュールファイル中の I D )  
 G – P C の対応を示す 1 文字 ( a, b, c, d ) または null  
       周波数グループ 1-4 または 0 に対応

## Type III

ape\_expid\_sid1sid2\_scanid\_YYYYDDDDHHMMSS.txt

ここで ‘ape’ – 固定文字  
 expid – 実験コード  
 sid1 – X 局 I D ( 小文字 2 文字 ) 大文字の場合は小文字に変換  
 sid2 – Y 局 I D ( 小文字 2 文字 ) 大文字の場合は小文字に変換  
 scanid – スキャン I D  
           VEX ファイル使用時はスキャン I D そのまま  
           SKED 使用時は ddd-hhmm 同じ分の中に複数の観測がある場合は 2 つ目以降に時間順に最後に a, b, c, d, ... をつけていく  
 YYYYDDDDHHMMSS – 観測開始時刻

## 補足 3: クロックパラメータとクロックエポック

クロックレートが 0 でない場合はエポックを設定することにより、スキャン毎のオフセットを自動的に計算する。一般的なクロックパラメータの決定は以下のように行う。  
 スキャン # 1 の PRT を  $t_1$ 、クロックオフセットを  $c_1$ 、最後のスキャンの PRT を  $t_e$ 、クロックオフセットを  $c_e$  とすると、クロックレートは  $(c_e - c_1)/(t_e - t_1)$  で得られる。apri\_calc を走らせる際のクロックオフセットとしては  $c_1$ 、レートは上記で求めた値、エポックは  $t_1$  を指定する。

## 3.1.2 実行例 ( 環境変数を確認・スケジュールファイルの中身確認 )

例 1 apri\_calc で使用する環境変数を確認する

```
$ apri_calc env
apri_calc (Ver. 2016-10-12)
  Enviroment variables
    K5SKED --- default directory for schedule file
                ( (null) ), program deflt is ( ./ )
    K5APRIDIR --- default directory for apriori file
                ( (null) ), program deflt is ( ../corrapri/ )
    K5NAMING --- default K5 file naming rule
                1 | -1 | 2 | -2 | 3 | 5
                ( (null) ), program deflt is ( 1 )
    K5SATKEY --- default satellite keyword of 1ch obs
```

```
( (null) ), program deflt is ( HYB NOZ GEO )
```

## 例2 スケジュールファイルの中身をチェックする

```
$ apri_calc -k10189.skd (または apri_calc k10189.skd -monit)
apri_calc (Ver. 2016-10-12)

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDDNNNN.dat (SKED deflt)
=====
SkdMonit: ***** Schedule file monitor *****
SkdMonit: Schedule file = ./k10189.skd
SkdMonit: Expcode = K10189
SkdMonit: Total Scan # = 2203
SkdMonit: Total Star # = 12
SkdMonit:
SkdMonit: 1st Scan = 2010/07/07 20:05:00 3C84
SkdMonit: Last Scan = 2010/07/11 00:00:21 3C84
SkdMonit:
SkdMonit: ----- Station ID Table -----
SkdMonit: G --- KOGANEI
SkdMonit: R --- KASHIM11
SkdMonit:
SkdMonit: ----- Star Table -----
SkdMonit:
SkdMonit: NAME1 NAME2 R.A.(deg) DEC(deg) EPOCH
SkdMonit: 1 0212+735 $ 34.378389 73.825728 2000.000000
SkdMonit: 2 0727-115 $ 112.579635 -11.686833 2000.000000
SkdMonit: 3 1921-293 $ 291.212733 -29.241700 2000.000000
SkdMonit: 4 2134+004 2134+00 324.160776 0.698393 2000.000000
SkdMonit: 5 2145+067 $ 327.022744 6.960723 2000.000000
SkdMonit: 6 1226+023 3C273B 187.277916 2.052388 2000.000000
SkdMonit: 7 1253-055 3C279 194.046527 -5.789312 2000.000000
SkdMonit: 8 1641+399 3C345 250.745042 39.810276 2000.000000
SkdMonit: 9 2223-052 3C446 336.446914 -4.950386 2000.000000
SkdMonit: 10 2251+158 3C454.3 343.490616 16.148211 2000.000000
SkdMonit: 11 0316+413 3C84 49.950667 41.511695 2000.000000
SkdMonit: 12 0923+392 4C39.25 141.762558 39.039126 2000.000000
SkdMonit:
SkdMonit: ----- Frequency (MHz) Table -----
SkdMonit: Gr# 1 7700.99 U 7710.99 U 7720.99 U 7850.99 U
SkdMonit: Gr# 2 8090.99 U 8290.99 U 8490.99 U 8550.99 U
SkdMonit: Gr# 3 8570.99 U 8580.99 U 2210.99 U 2220.99 U
SkdMonit: Gr# 4 2240.99 U 2290.99 U 2330.99 U 2340.99 U
SkdMonit:
SkdMonit: ----- PCAL Freq (kHz) Table -----
SkdMonit: Gr# 1 4010.0 4010.0 4010.0 4010.0
SkdMonit: Gr# 2 4010.0 4010.0 4010.0 4010.0
SkdMonit: Gr# 3 4010.0 4010.0 4010.0 4010.0
SkdMonit: Gr# 4 4010.0 4010.0 4010.0 4010.0
SkdMonit: -----
```

## 例3 スケジュールファイルの中身をチェックする(特定の電波源を指定する)

```
$ apri_calc -k10189.skd -source 3C273B <== '-source' オプションで抽出する観測の電波源
を 3C273B にセットする
apri_calc (Ver. 2016-10-12)

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDDNNNN.dat (SKED deflt)
Pickup Source Name : 3C273B <== 抽出するスキャンの電波源名がセットされる
=====
SkdMonit: ***** Schedule file monitor *****
SkdMonit: Schedule file = ./k10189.skd
SkdMonit: Expcode = K10189
SkdMonit: Total Scan # = 2203
SkdMonit: Total Star # = 12
SkdMonit:
SkdMonit: 1st Scan = 2010/07/07 20:05:00 3C84
SkdMonit: Last Scan = 2010/07/11 00:00:21 3C84
SkdMonit:
```



```

SkdMonit: ----- Station ID Table -----
SkdMonit:  G --- KOGANEI
SkdMonit:  R --- KASHIM11
SkdMonit: -----
SkdMonit: ----- Star Table -----
SkdMonit:      NAME1      NAME2  R.A.(deg)  DEC(deg)  EPOCH
SkdMonit:  1  0212+735      $  34.378389  73.825728  2000.000000
SkdMonit:  2  0727-115      $ 112.579635 -11.686833  2000.000000
SkdMonit:  3  1921-293      $ 291.212733 -29.241700  2000.000000
SkdMonit:  4  2134+004    2134+00 324.160776  0.698393  2000.000000
SkdMonit:  5  2145+067      $ 327.022744  6.960723  2000.000000
SkdMonit:  6  1226+023    3C273B 187.277916  2.052388  2000.000000
SkdMonit:  7  1253-055      3C279 194.046527 -5.789312  2000.000000
SkdMonit:  8  1641+399      3C345 250.745042 39.810276  2000.000000
SkdMonit:  9  2223-052      3C446 336.446914 -4.950386  2000.000000
SkdMonit: 10  2251+158    3C454.3 343.490616 16.148211  2000.000000
SkdMonit: 11  0316+413      3C84  49.950667 41.511695  2000.000000
SkdMonit: 12  0923+392    4C39.25 141.762558 39.039126  2000.000000
SkdMonit: -----
SkdMonit: ----- Frequency (MHz) Table -----
SkdMonit: Gr# 1  7700.99 U  7710.99 U  7720.99 U  7850.99 U
SkdMonit: Gr# 2  8090.99 U  8290.99 U  8490.99 U  8550.99 U
SkdMonit: Gr# 3  8570.99 U  8580.99 U  2210.99 U  2220.99 U
SkdMonit: Gr# 4  2240.99 U  2290.99 U  2330.99 U  2340.99 U
SkdMonit: -----
SkdMonit: ----- PCAL Freq (kHz) Table -----
SkdMonit: Gr# 1  4010.0  4010.0  4010.0  4010.0
SkdMonit: Gr# 2  4010.0  4010.0  4010.0  4010.0
SkdMonit: Gr# 3  4010.0  4010.0  4010.0  4010.0
SkdMonit: Gr# 4  4010.0  4010.0  4010.0  4010.0
SkdMonit: -----
SkdMonit: ----- PICKUP SCAN TABLE ----- <== 選択したスキャンが表示される
SkdMonit: SCAN#  SOURCE YYYY/DDD HH:MM:SS DURA  STATION_IDS
SkdMonit: スキャン番号 電波源 開始時刻 観測時間 参加局 ID
SkdMonit: 84  3C273B 2010/189 02:56:33 30  G R
SkdMonit: 87  3C273B 2010/189 03:01:02 30  G R
SkdMonit: 90  3C273B 2010/189 03:06:44 30  G R
SkdMonit: .... 中略 ....
SkdMonit: 1791 3C273B 2010/191 10:58:40 30  G R
SkdMonit: 1795 3C273B 2010/191 11:09:26 30  G R
SkdMonit: 1797 3C273B 2010/191 11:14:24 30  G R
SkdMonit: 1799 3C273B 2010/191 11:20:11 30  G R
SkdMonit: -----

```

このオプションを使うとフリンジサーチに使うスキャン番号を絞ることができる。基線が複数ある場合は '-baseid' オプションで基線を絞ることができる。

例 4 スケジュールファイルの中身をチェックする（特定の基線および電波源を指定する）

```

$ apri_calc -eg094a.vex -baseid KsSh -source 3C454.3
    <== 'baseid' で基線 ID(VEX ファイルの場合は局 ID は 2 文字なので基線 ID は 4 文字)
    '-source' オプションで抽出する観測の電波源を 3C454.3 にセットする

apri_calc (Ver. 2016-10-12)

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDDDNNNN.dat (SKED deflt)
Pickup Source Name : 3C454.3      <== セットされた電波源名
Pickup Baseline ID : KsSh         <== セットされた基線 ID
=====
search_site: No CLOCK info for ATCA included in the VEX FILE.
search_site: So all 0 for clock information was set.
search_site: No CLOCK info for PARKES included in the VEX FILE.
search_site: So all 0 for clock information was set.
.... 中略 ....
search_site: No CLOCK info for IRBENE included in the VEX FILE.
search_site: So all 0 for clock information was set.
SkdMonit: ***** Schedule file monitor *****
SkdMonit: Schedule file = ./eg094a.vex
SkdMonit: Expcode       = eg094a
SkdMonit: Total Scan #  = 74
SkdMonit: Total Star #  = 4
SkdMonit:

```



```

SkdMonit: 1st Scan = 2016/09/20 12:57:00 2223-052
SkdMonit: Last Scan = 2016/09/20 23:47:10 2215+020
SkdMonit:
SkdMonit: ----- Station ID Table -----
SkdMonit: At --- ATCA
SkdMonit: Pa --- PARKES
SkdMonit: Mp --- MOPRA
SkdMonit: Ho --- HOB_DBBC
SkdMonit: Cd --- CDDBBC
SkdMonit: Ti --- DSS43LBA
SkdMonit: Ks --- KASHIM34
SkdMonit: Ww --- WARK12M
SkdMonit: T6 --- TIANMA65
SkdMonit: Ur --- URUMQI
SkdMonit: Sh --- SHANGHAI
SkdMonit: Bd --- BADARY
SkdMonit: .... 中略 ....
SkdMonit: Mc --- MEDICINA
SkdMonit: O8 --- ONSALA85
SkdMonit: Tr --- TORUN
SkdMonit: Hh --- HART
SkdMonit: Ir --- IRBENE
SkdMonit: (Note that 1 char station ID is not defined in a VEX file)
SkdMonit: -----
SkdMonit: ----- Star Table -----
SkdMonit: NAME1 NAME2 R.A.(deg) DEC(deg) EPOCH
SkdMonit: 1 FAKERA 180.000000 85.000000 2000.000000
SkdMonit: 2 2215+020 334.450991 2.336309 2000.000000
SkdMonit: 3 2223-052 336.446914 -4.950386 2000.000000
SkdMonit: 4 3C454.3 343.490616 16.148211 2000.000000
SkdMonit: -----
SkdMonit: ----- Frequency (MHz) Table -----
SkdMonit: Gr# 1 1668.00 L 1668.00 U 1668.00 L 1668.00 U
SkdMonit: -----
SkdMonit: ----- PCAL Freq (kHz) Table -----
SkdMonit: Gr# 1 2000.0 2000.0 2000.0 2000.0
SkdMonit: -----
SkdMonit: ----- PICKUP SCAN TABLE ----- <==基線 ID, 電波源名の条件を満たすス
キャンが表示される
SkdMonit: SCAN# SOURCE YYYY/DDD HH:MM:SS DURA STATION_IDS
SkdMonit: 14 3C454.3 2016/264 14:41:30 480 At Pa Mp Ho Cd Ti Ks Ww T6 Ur Sh Bd
SkdMonit: 29 3C454.3 2016/264 17:01:00 480 At Mp Ho Cd Ti Ks T6 Ur Sh Bd Sv Zc
SkdMonit: -----

```

### 3.1.3 実行例 (K5/VSSP データ処理の場合)

例 1 非会話型で基線 ID=RG、観測番号 8 のみの予測値ファイルを作成。スケジュールファイル、データはカレントディレクトリにあるとし、予測値ファイルはカレントディレクトリに作成する。クロックオフセットは  $8\mu\text{sec}$  とする

```

$ apri_calc k10189.skd -baseid RG -apedir ./ -start 8 -stop 8 -coffset 8.0e-6
apri_calc (Ver. 2016-10-12)

```

```

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDNNNN.dat (SKED deflt)
Sub-net mode ON : PRT is set according to each scan length
Output directory : ./
=====

```

```

***** Schedule File Information *****
File name      --- ./k10189.skd
File type      --- SKED
Exp. code      --- K10189
# of stations  --- 2
                G R
# of stars     --- 12
# of scans     --- 2203
1st Scan : 2010/07/07 20:05:00 3C84
Last Scan : 2010/07/11 00:00:21 3C84
*****

```

```

----- STATION ID TABLE -----
G --- KOGANEI
R --- KASHIM11
-----
Station ID for X station : R
Station ID for Y station : G
Selected Baseline is R-G
Sampling Information is as follows
           from schedule file           from operator
KASHIM11 : 32MHz 1bit 16CH             0MHz 0bit 0CH
KOGANEI  : 32MHz 1bit 16CH             0MHz 0bit 0CH

Data directory for X station (KASHIM11) : ./
Data directory for Y station (KOGANEI) : ./
Frequency group and frequencies
Gr# 1 : 7700.99MHz U 7710.99MHz U 7720.99MHz U 7850.99MHz U
Gr# 2 : 8090.99MHz U 8290.99MHz U 8490.99MHz U 8550.99MHz U
Gr# 3 : 8570.99MHz U 8580.99MHz U 2210.99MHz U 2220.99MHz U
Gr# 4 : 2240.99MHz U 2290.99MHz U 2330.99MHz U 2340.99MHz U
Frequency Group# : 1
Gr# 1 is selected
PCAL freq (kHz) : 4010.0 4010.0 4010.0 4010.0
Clock offset and rate : 8e-06 0
Clock Epoch : 0/000 00:00:00
X Clock offset against UTC : 0.000000

ut1,wobbx,wobby : 0.000000 0.000000 0.000000

# of scans --- 2203
Scan range : 8 - 8

      8 2010189001135 4C39.25          0          0          0

Apriori File ( ./ape1880008RGa.txt ) created

Total # of a-priori files created is 1

```

## 例2 会話型で処理をする（以下の例の通り）

```

$ apri_calc k10189.skd -apedir apeout < スケジュールファイルと出力ディレクトリ
                                         (デフォルト以外の場合)を指定して走らせる
apri_calc (Ver. 2016-10-12)

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDDNNNN.dat (SKED deflt)
Sub-net mode ON : PRT is set according to each scan length
Output directory : apeout          <== このディレクトリが無い場合は警告を出してストップ
する
=====

***** Schedule File Information *****
File name      --- ./k10189.skd
File type      --- SKED
Exp. code      --- K10189
# of stations  --- 2
                G R
# of stars     --- 12
# of scans     --- 2203
1st Scan : 2010/07/07 20:05:00 3C84
Last Scan : 2010/07/11 00:00:21 3C84
*****

----- STATION ID TABLE -----
G --- KOGANEI
R --- KASHIM11
-----
Enter Station ID for X station ----> R < X局とする ID
Enter Station ID for Y station ----> G < Y局とする ID
Selected Baseline is R-G
Sampling Information is as follows
           from schedule file           from operator
KASHIM11 : 32MHz 1bit 16CH             0MHz 0bit 0CH

```

```

KOGANEI : 32MHz 1bit 16CH 0MHz 0bit 0CH

Enter data directory for X station (KASHIM11)
--> ./ < X局データのあるディレクトリを指定
Enter data directory for Y station (KOGANEI)
--> ./ < Y局データのあるディレクトリを指定
Data directory for X station (KASHIM11) : ./
Data directory for Y station (KOGANEI) : ./
Frequency group and frequencies
Gr# 1 : 7700.99MHz U 7710.99MHz U 7720.99MHz U 7850.99MHz U
Gr# 2 : 8090.99MHz U 8290.99MHz U 8490.99MHz U 8550.99MHz U
Gr# 3 : 8570.99MHz U 8580.99MHz U 2210.99MHz U 2220.99MHz U
Gr# 4 : 2240.99MHz U 2290.99MHz U 2330.99MHz U 2340.99MHz U
Enter Frequency Group# ----> 2 < 処理する周波数グループを選択
Frequency Group# : 2
Gr# 2 is selected
PCAL freq (kHz) : 4010.0 4010.0 4010.0 4010.0
Enter Clock Offset (sec) ----> 0.0 < クロックオフセット
(フリンジサーチ後は結果を反映すること)
Enter Clock Rate (s/s) ----> 0.0 < 通常0でOK。ここで0でない値を入力すると
以下のように入力するとエラーになる
Enter Clock Epoch (YYYY/DDD-HH:MM:SS|YYYY/MM/DD-HH:MM:SS|YYYYDDDDHHMMSS)
0 for each PRT (old style)
----> 2010/07/07-20:05:00 < エポックを入力する。ここで0を入力すると、
それぞれのPRTがエポックと見なされます。つまり、
スキャン毎のクロックオフセットは固定値となる。
Clock offset and rate : 0 0 < - もしレートが0でない場合、ここに反映される
Clock Epoch : 0/000 00:00:00 < - もしレートが0でない場合、ここに反映される
X Clock offset against UTC : 0.000000 < 会話型ではこのパラメータはセット
できません
Enter UT1-UTC (sec) ----> 0.0 < 不明な場合は0で可
Enter Wobb X (arcsec) ----> 0.0 < 不明な場合は0で可
Enter Wobb Y (arcsec) ----> 0.0 < 不明な場合は0で可

ut1,wobbx,wobby : 0.000000 0.000000 0.000000

# of scans --- 2203
Enter Start Scan number ----> 8 < 処理するスキャンの範囲(開始)
Enter Stop Scan number ----> 8 < 処理するスキャンの範囲(終了)
Scan range : 8 - 8

8 2010189001135 4C39.25 0 0 0

Apriori File ( apeout/ape1880008RGr.txt ) created

Total # of a-priori files created is 1

$

```

### 例3 電波源を限定して会話型で処理をする(以下の例の通り)

```

$ apri_calc k10189.skd -source 3C273B < オプション'-source'で電波源を3C273Bに限定
する
apri_calc (Ver. 2016-10-12)

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDDDNNNN.dat (SKED deflt)
Sub-net mode ON : PRT is set according to each scan length
Pickup Source Name : 3C273B <== 抽出する電波源名
Output directory : ../corrapri/
=====

***** Schedule File Information *****
File name --- ./k10189.skd
File type --- SKED
Exp. code --- K10189
# of stations --- 2
G R

```

```

# of stars    --- 12
# of scans    --- 2203
1st Scan : 2010/07/07 20:05:00 3C84
Last Scan : 2010/07/11 00:00:21 3C84
*****

----- STATION ID TABLE -----
G --- KOGANEI
R --- KASHIM11
-----

Enter Station ID for X station ----> R
Enter Station ID for Y station ----> G
Selected Baseline is R-G
Sampling Information is as follows
                from schedule file          from operator
KASHIM11 :    32MHz 1bit 16CH          0MHz 0bit 0CH
KOGANEI  :    32MHz 1bit 16CH          0MHz 0bit 0CH

Enter data directory for X station (KASHIM11)
--> ./
Enter data directory for Y station (KOGANEI)
--> ./
Data directory for X station (KASHIM11) : ./
Data directory for Y station (KOGANEI) : ./
Frequency group and frequencies
Gr# 1 :    7700.99MHz U    7710.99MHz U    7720.99MHz U    7850.99MHz U
Gr# 2 :    8090.99MHz U    8290.99MHz U    8490.99MHz U    8550.99MHz U
Gr# 3 :    8570.99MHz U    8580.99MHz U    2210.99MHz U    2220.99MHz U
Gr# 4 :    2240.99MHz U    2290.99MHz U    2330.99MHz U    2340.99MHz U
Enter Frequency Group# ----> 1
Frequency Group# : 1
Gr# 1 is selected
PCAL freq (kHz) : 4010.0 4010.0 4010.0 4010.0
Enter Clock Offset (sec) ----> 0
Enter Clock Rate (s/s) ----> 0
Clock offset and rate : 0 0
Clock Epoch : 0/000 00:00:00
X Clock offset against UTC : 0.000000
Enter UT1-UTC (sec) ----> 0
Enter Wobb X (arcsec) ----> 0
Enter Wobb Y (arcsec) ----> 0

ut1,wobbx,wobby : 0.000000 0.000000 0.000000

# of scans    --- 2203
Enter Start Scan number ----> 0      <== すべてのスキャンを対象とする時は0を入力
Enter Stop Scan number ----> 0      <== すべてのスキャンを対象とする時は0を入力
Scan range : 1 - 2203

84 2010189025648    3C273B          0          0          0
                                     <== スケジュールから 3C273B のみが抽出される

Apriori File ( ../corrapri/ape1880084RGa.txt ) created
87 2010189030117    3C273B          0          0          0

Apriori File ( ../corrapri/ape1880087RGa.txt ) created
90 2010189030659    3C273B          0          0          0

.... 中略 ....

Apriori File ( ../corrapri/ape1881797RGa.txt ) created
1799 2010191112026    3C273B          0          0          0

Apriori File ( ../corrapri/ape1881799RGa.txt ) created

Total # of a-priori files created is 168
$

```

## 例4 例3を非会話型で実行する

```

$ apri_calc k10189.skd ../corrapri RG 0 0 1 0 0 ./ ./ 0 0 0 1 0 0 3C273B
apri_calc (Ver. 2016-10-12)

```

```

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDDNNNN.dat (SKED deflt)
Sub-net mode OFF: PRT is set based on the longest scan
Pickup Source Name : 3C273B
Output directory : ../corrapri
=====

***** Schedule File Information *****
File name      --- ./k10189.skd
File type      --- SKED
Exp. code      --- K10189
# of stations  --- 2
                G R
# of stars     --- 12
# of scans     --- 2203
1st Scan : 2010/07/07 20:05:00 3C84
Last Scan : 2010/07/11 00:00:21 3C84
*****

----- STATION ID TABLE -----
G --- KOGANEI
R --- KASHIM11
-----

Station ID for X station : R
Station ID for Y station : G
Selected Baseline is R-G
Sampling Information is as follows
                from schedule file                from operator
KASHIM11 : 32MHz 1bit 16CH                0MHz 0bit 0CH
KOGANEI  : 32MHz 1bit 16CH                0MHz 0bit 0CH

Data directory for X station (KASHIM11) : ./
Data directory for Y station (KOGANEI) : ./
Frequency group and frequencies
Gr# 1 : 7700.99MHz U 7710.99MHz U 7720.99MHz U 7850.99MHz U
Gr# 2 : 8090.99MHz U 8290.99MHz U 8490.99MHz U 8550.99MHz U
Gr# 3 : 8570.99MHz U 8580.99MHz U 2210.99MHz U 2220.99MHz U
Gr# 4 : 2240.99MHz U 2290.99MHz U 2330.99MHz U 2340.99MHz U
Frequency Group# : 1
Gr# 1 is selected
PCAL freq (kHz) : 4010.0 4010.0 4010.0 4010.0
Clock offset and rate : 0 0
Clock Epoch : 0/000 00:00:00
X Clock offset against UTC : 0.000000

ut1,wobbx,wobby : 0.000000 0.000000 0.000000

# of scans      --- 2203
Scan range : 1 - 2203

84 2010189025648 3C273B 0 0 0

Apriori File ( ../corrapri/ape1880084RGa.txt ) created
87 2010189030117 3C273B 0 0 0

.... 中略 ....

Apriori File ( ../corrapri/ape1881797RGa.txt ) created
1799 2010191112026 3C273B 0 0 0

Apriori File ( ../corrapri/ape1881799RGa.txt ) created

Total # of a-priori files created is 168
$

```

これで、予測値ファイルが apeout ディレクトリ以下にできる（出力ディレクトリを指定しない場合は../corrapri/ 以下にできる）。  
 予測値ファイル名は Type 1 の場合 apeDDDNNNNXYG.txt

ここで DDD - 通日  
 NNNN - 観測番号

```

XY      - 基線 I D
G       - 周波数グループ ('a','b','c','d') 1 - 4 に対応

```

### 3.1.4 実行例 (VDIF フォーマットデータ同志の相関処理の場合)

```

$ apri_calc ../sked/ks15002.skd -format VDIF < - - "-format"オプションで VDIF を指定する
apri_calc (Ver. 2016-10-12)

```

```

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDNNNN.dat (SKED deflt)
Sub-net mode ON : PRT is set according to each scan length
X station data format : VDIF < 選択データフォーマットの表示
Y station data format : VDIF < 選択データフォーマットの表示
Output directory : ../corrapri/
=====

```

```

***** Schedule File Information *****
File name --- ../src/ks15002.skd
File type --- SKED
Exp. code --- KS15002
# of stations --- 5
               G R Y B O
# of stars --- 16
# of scans --- 289
1st Scan : 2015/01/02 02:00:00 3C345
Last Scan : 2015/01/03 02:02:43 0059+581
*****

```

```

----- STATION ID TABLE -----

```

```

G --- KOGANEI
R --- KASHIM11
Y --- TATEYAMA
B --- MIURA
O --- KASHIM34

```

```

Enter Station ID for X station ----> R

```

```

Enter Station ID for Y station ----> G

```

```

Selected Baseline is R-G

```

```

Sampling Information is as follows (サンプリング情報の表示)

```

```

               from schedule file      from operator
KASHIM11 : 16MHz 1bit 16CH      0MHz 0bit 0CH
KOGANEI : 16MHz 1bit 16CH      0MHz 0bit 0CH

```

```

Enter data directory for X station (KASHIM11)

```

```

----> ./ < データディレクトリの指定

```

```

Enter data directory for Y station (KOGANEI)

```

```

----> ./ < データディレクトリの指定

```

```

Data directory for X station (KASHIM11) : ./

```

```

Data directory for Y station (KOGANEI) : ./

```

```

Frequency group and frequencies

```

```

Gr# 1 : 7864.99MHz U 7874.99MHz U 7884.99MHz U 8014.99MHz U
Gr# 2 : 8114.99MHz U 8244.99MHz U 8504.99MHz U 8544.99MHz U
Gr# 3 : 8564.99MHz U 8574.99MHz U 2214.99MHz U 2224.99MHz U
Gr# 4 : 2234.99MHz U 2264.99MHz U 2294.99MHz U 2304.99MHz U

```

```

Enter Frequency Group# (0 for all) ----> 0 < すべての選択

```

```

All frequency mode is selected!

```

```

Enter Clock Offset (sec) ----> 0

```

```

Enter Clock Rate (s/s) ----> 0

```

```

Clock offset and rate : 0 0

```

```

Clock Epoch : 0/000 00:00:00

```

```

X Clock offset against UTC : 0.000000

```

```

Enter UT1-UTC (sec) ----> 0

```

```

Enter Wobb X (arcsec) ----> 0

```

```

Enter Wobb Y (arcsec) ----> 0

```

```

ut1,wobbx,wobby : 0.000000 0.000000 0.000000

```

```

# of scans --- 289

```

```

Enter Start Scan number ----> 1

```

```

Enter Stop Scan number ----> 2

```

```

Scan range : 1 - 2

```

```

1 2015002020045    3C345                0                0                0
Apriori File ( ../corrapri/ape0020001RG.txt ) created
2 2015002020520    3C454.3              0                0                0
Apriori File ( ../corrapri/ape0020002RG.txt ) created
Total # of a-priori files created is 2
$

```

必要に応じて出来上がった予測値ファイル中のデータファイル名をテキストエディタで修正する

### 3.1.5 実行例 (Mark-5B フォーマットデータ同志の相関処理の場合)

```
$ apri_calc ../sked/ks15002.skd -format M5B < - - "-format"オプションで M5B を指定する
apri_calc (Ver. 2016-10-12)
```

```

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDNNNN.dat (SKED deflt)
Sub-net mode ON : PRT is set according to each scan length
X station data format :   Mark-5B          <   選択データフォーマットの表示
Y station data format :   Mark-5B          <   選択データフォーマットの表示
=====

```

```

***** Schedule File Information *****
File name      --- ../src/ks15002.skd
File type      --- SKED
Exp. code      --- KS15002
# of stations  --- 5
                  G R Y B 0
# of stars     --- 16
# of scans     --- 289
1st Scan : 2015/01/02 02:00:00 3C345
Last Scan : 2015/01/03 02:02:43 0059+581
*****

```

```

----- STATION ID TABLE -----
G --- KOGANEI
R --- KASHIM11
Y --- TATEYAMA
B --- MIURA
O --- KASHIM34
-----

```

```

Enter Station ID for X station ----> R
Enter Station ID for Y station ----> G
Selected Baseline is R-G
Sampling Information is as follows      ( サンプルング情報の表示 )
                from schedule file      from operator
KASHIM11 :    16MHz 1bit 16CH          0MHz 0bit 0CH
KOGANEI   :    16MHz 1bit 16CH          0MHz 0bit 0CH

```

```

Enter data directory for X station (KASHIM11)
---> ./          <   データディレクトリの指定
Enter data directory for Y station (KOGANEI)
---> ./          <   データディレクトリの指定
Data directory for X station (KASHIM11) : ./
Data directory for Y station (KOGANEI) : ./
Frequency group and frequencies
Gr# 1 : 7864.99MHz U 7874.99MHz U 7884.99MHz U 8014.99MHz U
Gr# 2 : 8114.99MHz U 8244.99MHz U 8504.99MHz U 8544.99MHz U
Gr# 3 : 8564.99MHz U 8574.99MHz U 2214.99MHz U 2224.99MHz U
Gr# 4 : 2234.99MHz U 2264.99MHz U 2294.99MHz U 2304.99MHz U
Enter Frequency Group# (0 for all) ----> 0 <   すべての選択
All frequency mode is selected!
Enter Clock Offset (sec) ----> 0
Enter Clock Rate (s/s) ----> 0
Clock offset and rate : 0 0
Clock Epoch : 0/000 00:00:00
X Clock offset against UTC : 0.000000
Enter UT1-UTC (sec) ----> 0

```



```

Enter Wobb X (arcsec) ----> 0
Enter Wobb Y (arcsec) ----> 0

ut1,wobbx,wobby : 0.000000 0.000000 0.000000

# of scans    --- 289
Enter Start Scan number ----> 1
Enter Stop Scan number ----> 2
Scan range : 1 - 2

1 2015002020045    3C345                0                0                0

Apriori File ( ../corrapri/ape0020001RG.txt ) created
2 2015002020520    3C454.3            0                0                0

Apriori File ( ../corrapri/ape0020002RG.txt ) created

Total # of a-priori files created is 2

$

```

必要に応じて出来上がった予測値ファイル中のデータファイル名をテキストエディタで修正する

### 3.1.6 実行例 ( X 局データが VSSP32、 Y 局データが VDIFF フォーマットデータの場合 )

\$ apri\_calc ../sked/ks15002.skd -formY VDIFF < - - "-formY" オプションで VDIFF を指定する。  
X 局はデフォルトが VSSP フォーマットなので指定する必要はない。

apri\_calc (Ver. 2016-10-12)

```

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDNNNN.dat (SKED deflt)
Sub-net mode ON : PRT is set according to each scan length

Y station data format :    VDIFF                <    選択データフォーマットの表示
=====

***** Schedule File Information *****
File name      --- ../src/ks15002.skd
File type      --- SKED
Exp. code      --- KS15002
# of stations  --- 5
                  G R Y B O
# of stars     --- 16
# of scans     --- 289
1st Scan : 2015/01/02 02:00:00 3C345
Last Scan : 2015/01/03 02:02:43 0059+581
*****

----- STATION ID TABLE -----
G --- KOGANEI
R --- KASHIM11
Y --- TATEYAMA
B --- MIURA
O --- KASHIM34
-----

Enter Station ID for X station ----> R
Enter Station ID for Y station ----> G
Selected Baseline is R-G
Sampling Information is as follows      ( サンプル情報の表示 )
              from schedule file          from operator
KASHIM11 :    16MHz 1bit 16CH          0MHz 0bit 0CH
KOGANEI  :    16MHz 1bit 16CH          0MHz 0bit 0CH

Enter data directory for X station (KASHIM11)
---> ./                <    データディレクトリの指定
Enter data directory for Y station (KOGANEI)
---> ./                <    データディレクトリの指定
Data directory for X station (KASHIM11) : ./
Data directory for Y station (KOGANEI) : ./
Frequency group and frequencies
Gr# 1 :    7864.99MHz U 7874.99MHz U 7884.99MHz U 8014.99MHz U

```



```

Gr# 2 : 8114.99MHz U 8244.99MHz U 8504.99MHz U 8544.99MHz U
Gr# 3 : 8564.99MHz U 8574.99MHz U 2214.99MHz U 2224.99MHz U
Gr# 4 : 2234.99MHz U 2264.99MHz U 2294.99MHz U 2304.99MHz U
Enter Frequency Group# ----> 1 < どれか一つのグループを選択
All frequency mode is selected!
Enter Clock Offset (sec) ----> 0
Enter Clock Rate (s/s) ----> 0
Clock offset and rate : 0 0
Clock Epoch : 0/000 00:00:00
X Clock offset against UTC : 0.000000
Enter UT1-UTC (sec) ----> 0
Enter Wobb X (arcsec) ----> 0
Enter Wobb Y (arcsec) ----> 0

```

```

ut1,wobbx,wobby : 0.000000 0.000000 0.000000

```

```

# of scans --- 289
Enter Start Scan number ----> 1
Enter Stop Scan number ----> 2
Scan range : 1 - 2

```

```

1 2015002020045 3C345 0 0 0

```

```

Apriori File ( ../corrapri/ape0020001RG.txt ) created
2 2015002020520 3C454.3 0 0 0

```

```

Apriori File ( ../corrapri/ape0020002RG.txt ) created

```

```

Total # of a-priori files created is 2

```

```

$

```

必要に応じて出来上がった予測値ファイル中のデータファイル名をテキストエディタで修正する

### 3.1.7 実行例 (X局、Y局ともに VDIF データ、ただし Y 局データは単 CH/スレッドのマルチスレッドデータの場合)

```

$ apri_calc ../sked/q19311.skd -apedir ../work -formX VDIF 16M2B8C -formY VDIF 16M2B1C
  -threadY MULTI

```

```

apri_calc (Ver. 2019-12-03)

```

```

===== RUN CONDITION =====
K5 file naming type is Type 1 : sidDDNNNN.dat (SKED deflt)
Sub-net mode ON : PRT is set according to each scan length
X station data format : VDIF
Y station data format : VDIF
Output directory : ../work
=====

```

```

***** Schedule File Information *****

```

```

File name --- ../sked/q19311.skd
File type --- SKED
Exp. code --- Q19311
# of stations --- 2
               M V
# of stars --- 88
# of scans --- 28
1st Scan : 2019/11/07 10:00:00 2229+695
Last Scan : 2019/11/07 10:59:15 0805+410

```

```

*****

```

```

----- STATION ID TABLE -----

```

```

M --- MK-VLBA
V --- WETTZELL

```

```

-----
Enter Station ID for X station ----> M

```

```

Enter Station ID for Y station ----> V

```

```

Selected Baseline is M-V

```

```

Sampling Information is as follows

```

	from schedule file	from operator
MK-VLBA :	16MHz 2bit 8CH	16MHz 2bit 8CH
WETTZELL :	16MHz 2bit 8CH	16MHz 2bit 1CH

```

Enter data directory for X station (MK-VLBA)
----> ./
Enter data directory for Y station (WETTZELL)
----> ./
Data directory for X station (MK-VLBA) : ./
Data directory for Y station (WETTZELL) : ./
Frequency group, frequency, sideband, polarization based on MK-VLBA are
Gr# 1 :      8365.75 MHz U      8445.75 MHz U      8805.75 MHz U      8925.75 MHz U
Gr# 2 :      2232.75 MHz U      2262.75 MHz U      2372.75 MHz U      2392.75 MHz U
Select Frequency Group#(s) (e.g. 1 [2 3 4 ...]) or 0 for all ----> 0
All frequency mode is selected!
Enter Clock Offset (sec) ----> 0
Enter Clock Rate (s/s) ----> 0
Clock offset and rate : 0 0
Clock Epoch : 0/000 00:00:00
X Clock offset against UTC : 0
Enter UT1-UTC (sec) ----> 0
Enter Wobb X (arcsec) ----> 0
Enter Wobb Y (arcsec) ----> 0

ut1,wobbx,wobby : 0.000000 0.000000 0.000000

# of scans --- 28
Enter Start Scan number ----> 1
Enter Stop Scan number ----> 1
Scan range : 1 - 1

1 2019311100020 2229+695          0          0          0

Apriori File ( ../work/ape3110001MV.txt ) created

Total # of a-priori files created is 1

```

作成される予測値ファイルの中身は以下のようになる

```

** This is Apriori file made by apri_calc Ver. 2019-12-03
**   for cor, cor_all, fx_cor, and fx_cor_all
**
** (X clock offset wrt UTC is not reflected to a-priori values)
**
** SUBNET ON: PRT is set according to each scan length
**
** Clock parameters at run are as follows,
**   Clock Offset (s) : 0.000000
**   Clock Rate (s/s) : 0.000000
**   Clock Epoch      :      0000/000 00:00:00
**
**
$EXPCODE
Q19311

$OBS_NUMBER
1

$STATION1
MK-VLBA ./M3110001

$FORMAT1
VDIF 16MHz 8CH 2bit

$XYZ-STATION1
-5464074.952510 -2495249.107380 2148296.843470

$STATION2
WETTZELL ./V3110001

$FORMAT2
VDIF 16MHz 1CH 2bit          <= 単 CH/スレッドデータなので CH 数は 1 であることに注意！

$XYZ-STATION2
4075539.899410 931735.270250 4801629.351850

$BASEID

```

MV

\$FRQ\_GRP(1-2)  
0

\$FREQUENCY

8365750000.0 U 1 1 -- (0-0)  
8445750000.0 U 2 2 -- (0-1)  
8805750000.0 U 3 3 -- (0-2)  
8925750000.0 U 4 4 -- (0-3)  
2232750000.0 U 5 5 -- (0-4)  
2262750000.0 U 6 6 -- (0-5)  
2372750000.0 U 7 7 -- (0-6)  
2392750000.0 U 8 8 -- (0-7)

<== ( ) の中が X,Y 局のスレッド#

\$PCAL\_FREQ

10000.0  
10000.0  
10000.0  
10000.0  
10000.0  
10000.0  
10000.0  
10000.0

\$CLOCK

OFST= 0.000000  
RATE= 0.000000  
XCOF= 0.000000

\$SOURCE

2229+695

\$RA

22 30 36.46970500

\$DEC

69 46 28.07690000

\$EPOCH

2000.0

\$GHA

14 34 58.530000

\$EOP

UT1-UTC= 0.000000  
X\_WOBB = 0.000000  
Y\_WOBB = 0.000000

\$START

2019311100000

\$STOP

2019311100040

\$APRIORI

PRT=2019311100020  
TAU0= -2.210227274619678e-03  
TAU1= -7.220902109727506e-07  
TAU2= -3.243827760964927e-11  
TAU3= 3.811317364141309e-15

\$END

## 3.2 skdchk

ユーティリティ名

skdchk

機能

スケジュールファイルのチェック

スケジュールファイルを読み、局情報、電波星情報、観測情報を出力する。最後に、全観測時間を秒で示すと共に、必要なディスク容量を局毎に表示する。

Ver.2019-12-21 以降では SKED ファイルの \$CODES ブロックの周波数と CH# の対応解析方法を改良し、各局毎の周波数情報を厳密に解析している。旧方式との互換性を保つためオプション ‘-LEGACY’ を用意。

### 3.2.1 実行方法

```
skdchk sked_file [-NOEARLY] [-LEGACY]
```

ここで sked\_file        -    スケジュールファイル名

-NOEARLY        -    “tape early start” パラメータを無視する

-LEGACY        -    SKED ファイルの \$CODES ブロックの周波数と CH# の対応解析を  
旧方式で行う ( Ver.2019-12-21 以前との互換性のため )

### 3.2.2 実行例

```
$ skdchk jd0306.skd
skdchk Ver 2.31        2016-10-12
Schedule file is    jd0306.skd

*** SCHEDULE FILE (jd0306.skd) INFORMATION ***
Experiment code     : JD0306
Number of stations : 8
 1 T    TSUKUB32    -3957408.751200    3310229.346600    3737494.836000
 2 A        AIRA    -3530219.322300    4118797.541900    3344015.905900
 3 C    CHICHI10   -4490618.469200    3483908.166600    2884899.205700
 4 J    SINTOTU3   -3642141.844800    2861496.642500    4370361.717900
 5 R    KASHIM11   -3997505.701700    3276878.404550    3724240.703140
 6 Y        GIFU11   -3787123.360830    3564181.693760    3680274.907440
 7 H    TOMAKO11   -3680586.301730    2917515.745560    4300987.652680
 8 K    YAMAGU32   -3502535.908490    3950950.219310    3566374.002980
Number of stars : 114
(only 20 stars are listed here)
 1    0003-066        $    1.557887    -6.393149    2000.000000
 2    0014+813        $    4.285312    81.585593    2000.000000
 3    0048-097        $    12.672156    -9.484781    2000.000000
 4    0059+581        $    15.690677    58.403094    2000.000000
 5    0104-408        $    16.687950   -40.572211    2000.000000
 6    0106+013        $    17.161546     1.583421    2000.000000
 7    0111+021        $    18.429771     2.371477    2000.000000
 8    0119+115        $    20.423313    11.830670    2000.000000
 9    0119+041        $    20.486924     4.373537    2000.000000
10    0133+476        $    24.244145    47.858083    2000.000000
11    0201+113        $    30.944404    11.579280    2000.000000
12    0202+149        $    31.210058    15.236401    2000.000000
13    0208-512        $    32.692502   -51.017192    2000.000000
14    0229+131        $    37.941225    13.381866    2000.000000
15    0235+164        $    39.662209    16.616465    2000.000000
16    0316+413        3C84    49.950667    41.511695    2000.000000
17    0336-019        CTA26    54.878907    -1.776612    2000.000000
18    0355+508        NRA0150    59.873947    50.963934    2000.000000
19    0402-362        $    60.973958   -36.083865    2000.000000
20    0405-385        $    61.745981   -38.441123    2000.000000
Number of scans : 209
First 5 scans are as follows:
 1        CTA26    3/197 02:00:00    310
 2    1803+784    3/197 02:06:10    784
 3        4C39.25    3/197 02:19:50    100
 4    0727-115    3/197 02:23:10    180
 5    0537-441    3/197 02:26:40    190
Last 5 scans are as follows:
205    1044+719    3/198 01:22:20    430
206    0552+398    3/198 01:33:30    160
207    0202+149    3/198 01:37:00    784
208    0133+476    3/198 01:50:30    550
209    0537-441    3/198 02:00:40    180
```

Early Start Parameter (sec) : 0

Maximum Disk requirements

Total observation time (sec) = 76038

Disk requirements

32Mbps : 304.152 GBytes  
 64Mbps : 608.304 GBytes  
 128Mbps : 1216.608 GBytes  
 256Mbps : 2433.216 GBytes

Disk Requirements by Station (GBytes)

	TSUKUB32	AIRA	CHICHI10	SINTOTU3	KASHIM11	GIFU11	TOMAKO11
sec	46018	65066	65976	37808	76038	76038	76038
#scans	200	186	189	169	209	209	209
32Mbps	184.1	260.3	263.9	151.2	304.2	304.2	304.2
64Mbps	368.1	520.5	527.8	302.5	608.3	608.3	608.3
128Mbps	736.3	1041.1	1055.6	604.9	1216.6	1216.6	1216.6
256Mbps	1472.6	2082.1	2111.2	1209.9	2433.2	2433.2	2433.2

Disk Requirements by Station (GBytes)

	YAMAGU32
sec	29012
#scans	78
32Mbps	116.0
64Mbps	232.1
128Mbps	464.2
256Mbps	928.4

## 4 相関処理関係ソフトの使い方

### 4.1 `fx_cor`, `fx_cor_new`, `cor`, `cor_new`

#### ユーティリティ名

`fx_cor`, `fx_cor_new`, `cor`, `cor_new`

#### 機能

予測値ファイルを読み、書かれている情報に従って相関処理を実行する。

`fx_cor` および `fx_cor_new` は FX 方式の相関処理ソフトウェアであり、`cor` および `cor_new` は 1 ビットサンプリングデータに特化した XF 方式の相関処理ソフトウェアである。1 ビットサンプリングで観測された通常の測地実験処理（ラグ数 32）では `cor` および `cor_new` による処理の方が `fx_cor` や `fx_cor_new` より高速である。

なお最後に ‘\_new’ が付くソフトウェアは K5/VSSP フォーマットデータ以外に VDIF, Mark5B, ADS, OCTAD, RAW 形式データの処理（含む異種形式データ間の相関処理）が可能である。

#### 4.1.1 実行方法

##### 【方法 1】 通常の相関処理

`fx_cor` の場合について説明するがそれ以外のソフトの場合は `fx_cor` のところをそのソフトウェア名に置き換える。

`fx_cor afile [options]`

ここで <i>afile</i>	–	アプリオリファイル名 0 とすると内部のデフォルトファイル名を使用
<i>options</i> (順不同)		
<code>-integ integration_time</code>	–	総積分時間 (sec) 負の値は可能な限りの積分時間 デフォルトまたは 0 はアプリオリファイルに記述している観測時間
<code>-coffset clock_offset</code>	–	クロックオフセット (sec) (Y 局が進んでいる場合を正) デフォルトは 0.0
<code>-crate clock_rate</code>	–	クロックレートオフセット (s/s) デフォルトは 0.0
<code>-soffset start_offset</code>	–	開始時刻のオフセット (整数秒単位) デフォルトは 0
<code>-tlpp tlpp</code>	–	単位積分時間 (PP) の設定 (sec) (デフォルトは 1.0) 1 秒以下の値を指定するときは、1 秒を整数で割った値とすること (例: 0.2)
<code>-smode smode</code>	–	一度に処理するサンプル数 (遅延サーチ範囲) モード (プログラマー用) 0: 最大 200000 点の範囲の相関関数 (低速) 1: 中 10000 点の範囲サーチ 2: 最小 1000 (または 2000) 点の範囲サーチ (高速) デフォルトは 2

- pp\_nosync                    -    PP 同期モードを非同期にセット ( 処理開始可能な時刻から PP を開始 )  
デフォルトは PP の開始が秒に同期する
  - lag *delsize*                -    ラグウィンドウサイズのセット。  $2^n$  の値をセットする  
例 : 16, 32, 64, 128, 256, 512, 1024, 2048, ....  
0 の場合はプログラム中の DELAYSIZE で示される値  
デフォルトは 0 ( 通常 DELAYSIZE は 32 )
  - pmode *pmode*                -    プロット表示デバイスモード  
0: XWINDOW および PostScript ファイル (pgplot.ps または gnuplot.ps) 出力 ( デフォルト )  
1: PostScript ファイル (pgplot.ps または gnuplot.ps) 出力のみ  
2: XWINDOW 出力のみ  
-1: グラフ出力無し
  - comment "*any comment*"                -    コメント ( pmode=0,1,2 の場合グラフ上部に表示 )
  - nopcal                        -    P C A L 検出機能を抑制する
  - ch1 *ch1Y*                    -    X 局の CH1 と相関処理を行う Y 局の CH を指定 ( デフォルトは 1 )
  - ch2 *ch2Y*                    -    X 局の CH2 と相関処理を行う Y 局の CH を指定 ( デフォルトは 2 )
  - ch3 *ch3Y*                    -    X 局の CH3 と相関処理を行う Y 局の CH を指定 ( デフォルトは 3 )
  - ch4 *ch4Y*                    -    X 局の CH4 と相関処理を行う Y 局の CH を指定 ( デフォルトは 4 )
- 注 *fx\_cor\_new* および *cor\_new* の場合は -ch16 *ch16Y* までセット可能
- orule *naming\_rule*                -    相関出力ファイルの命名則を指定  
0 : プログラム固定ファイル名 (cout.txt)  
1 : coutNNNN.txt (デフォルト)  
    ここで NNNN : 4 桁の通し番号  
2 : coutEXP\_CODE/coutYYDDDNNNNXYG.txt  
    ここで    EXP\_CODE    -    実験コード  
              YY            -    年 ( 2 桁 )  
              DDD          -    通日 ( 3 桁 )  
              NNNN        -    スキャン番号 ( 4 桁 )  
              XY          -    基線 I D ( 2 文字または 4 文字 )  
              G            -    周波数グループ (a|b|c|d) または null  
3 : coutYYDDDNNNNXYG.txt  
    ここで    YY          -    年 ( 2 桁 )  
              DDD          -    通日 ( 3 桁 )  
              NNNN        -    スキャン番号 ( 4 桁 )  
              XY          -    基線 I D ( 2 文字または 4 文字 )  
              G            -    周波数グループ (a|b|c|d) または null
- 注 : *cor* および *cor\_new* 処理では出力ファイル名中 "cout" の部分が "coutt" となる
- odir *outdir*                -    相関出力ファイル作成ディレクトリを指定する  
デフォルトは環境変数 K5COUT で指定されるディレクトリ、環境変数が指定されていない場合は、../cout ディレクトリ
  - firstep *frstep*                -    フリンジ位相計算ステップの設定

- 0    -   自動設定
- 1    -   1 サンプル毎
- 8    -   8 サンプル毎 (デフォルト)
- $N$    -    $N$  サンプル毎 ( $1000/N$  は整数であること)
- frauto           -   フリンジ位相計算ステップを自動設定モードに設定 (“-firststep 0” と同じ)
- rffoffset *rf\_offset*   -   X 局と Y 局間の RF 周波数の差を設定 (RF<sub>y</sub>-RF<sub>x</sub>)(Hz)
- cout *cout\_file*       -   相関出力ファイル名を強制的に設定する
- fall *pcalf*           -   全 CH の PCAL 周波数を強制的に設定する (kHz)
- f1 *pcalf1*           -   CH1 の PCAL 周波数を強制的に設定する (kHz)
- f2 *pcalf2*           -   CH2 の PCAL 周波数を強制的に設定する (kHz)
- f3 *pcalf3*           -   CH3 の PCAL 周波数を強制的に設定する (kHz)
- f4 *pcalf4*           -   CH4 の PCAL 周波数を強制的に設定する (kHz)
- 注 *fx\_cor\_new* および *cor\_new* の場合は -f16 *pcalf16* までセット可能
- 1ch *x\_ch* [*y\_ch*]       -   1CH 処理モードの設定
  - x\_ch*   -   X 局チャンネル#
  - y\_ch*   -   Y 局チャンネル#
- === 以下は *fx\_cor* と *fx\_cor\_new* のみのオプション ===
- modefr *modefr*       -   フリンジストッピング関数近似モードの選択 (*cor* および *cor\_new* では無効)
  - 0    -   近似なし
  - 9    -   9 レベル近似 (デフォルト)
  - 2    -   2 レベル近似
  - 3    -   3 レベル近似
- hanning           -   ハニング窓関数を使用する (デフォルトは箱型窓関数)
- hamming           -   ハミング窓関数を使用する (デフォルトは箱型窓関数)
- bpf *flow:fhhigh[:fact][,flow:fhhigh[:fact][,flow:fhhigh[:fact][,.....]]]*
  - BPF (バンドパスフィルター) のセット (低域および高域カットオフ周波数による設定: 最大 20 個)
    - flow*   -   ベースバンドでの低域カットオフ周波数 (MHz)
    - fhhigh* -   ベースバンドでの高域カットオフ周波数 (MHz)
    - fact*   -   強度ファクター (0.0–1.0) デフォルトは 1.0
- bpf2 *fc:bw[:fact][,fc:[bw]:fact][,fc:[bw]:fact][,.....]]*
  - BPF (バンドパスフィルター) のセット (中心周波数と通過帯域幅による設定: 最大 20 個)
    - fc*    -   ベースバンドでの BPF の中心周波数 (MHz)
    - bw*    -   通過帯域幅 (MHz)。最初の BPF と同じ場合は省略可
    - fact*   -   強度ファクター (0.0–1.0) デフォルトは 1.0
- fres *fres*           -   BPF 処理時の周波数分解能の設定 (MHz)。デフォルトは自動設定
- delpcal *ch#:fs[:fi][,ch#:fs[:fi][,.....]]*
  - 除去する PCAL 信号の周波数をチャンネル毎にセット



*ch#* - チャンネル番号 (1 から開始)

*fs* - PCAL 信号周波数 (MHz)

*fi* - PCAL 信号の間隔 (MHz)

除去される PCAL 信号周波数は

$fs, fs + fi, fs + 2fi, fs + 3fi, \dots$

*fi* を省略した場合は *fs* のみが除去される

除去される帯域幅は FFT の分解周波数幅の 2 倍。

変更する場合には以下の '-pcalbw' オプションを使用する

または

-delpcal *delpcal\_file* - 除去する PCAL 信号情報を記述したファイルを指定する  
 -pcalbw *bwmhz* - PCAL 信号除去時に帯域幅をデフォルト値以外に設定 (MHz)  
 デフォルト値は FFT 分解周波数幅の 2 倍

=== 以下は fx\_cor\_new のみのオプション ===

-e2bit *mode* - X、Y データの 2-bit AD データエンコーディング法をセットする  
 0 - デフォルト  
 1 - (0, 1, 2, 3) オフセット 2 進数 (VDIF のデフォルト)  
 2 - (0, 2, 1, 3) Mark-5B のデフォルト  
 3 - (1, 0, 2, 3) 符号付き整数

注: このオプションが有効なのは VDIF または Mark-5B データのみである

-e2bitx *mode* - X データの 2-bit AD エンコーディング法をセットする  
 -e2bity *mode* - Y データの 2-bit AD エンコーディング法をセットする

オプション '-delpcal *delpcal\_file*' で指定するファイルの中身の例を以下に示す。

```
*** PCAL signal rejection filter file ***
**   for option '-delpcal' of fx_cor_new
**
**   usage   -delpcal DELPCAL.txt
**
**   Format Ver. 2019-08-27
**
**   ch#   sFreq(MHz) intFreq(MHz)
**   where
**   ch#   :   channel # (starting from 1)
**           0 means for all channel
**   sFreq :   lowest PCAL frequency in channel (MHz)
**   intFreq :   interval of PCAL signal (MHz)
**   Note-- rejected PCAL frequency is sFreq, sFreq+intFreq, sFreq+2*intFreq, ....
**   [example]
**   ch#   sFreq(MHz)   intFreq (MHz)
**   0     0.01         5.0
**   set for all channel same parameter
**
**   or
**   1     1.4           5.0
**   2     8.4           5.0
**   3     2.4           5.0
**   4     4.4           5.0
**   set different start frequency for every channel
**
**
* CH#   sFreq(MHz)   intFreq(MHz)
* 1     0.4           5.0
* 2     3.4           5.0
* 3     4.4           5.0
* 4     0.4           5.0
* 5     4.4           5.0
```

	6	1.4	5.0
	7	4.4	5.0
	8	2.4	5.0
*			

単位積分時間 ( $t1pp$ ) として実際に許される 1 秒以下の値は以下の表の通り。

サンプリング周波数	1pp 時間 (sec)						
	0.01	0.02	0.04	0.05	0.1	0.2	0.5
40kHz	×	OK	OK	×	OK	OK	OK
100kHz	×	×	OK	×	×	OK	×
200kHz	×	OK	OK	×	OK	OK	OK
500kHz	×	×	OK	×	×	OK	×
1MHz	×	OK	OK	×	OK	OK	OK
2MHz	OK	OK	OK	OK	OK	OK	OK
4MHz	OK	OK	OK	OK	OK	OK	OK
8MHz	OK	OK	OK	OK	OK	OK	OK
16MHz	OK	OK	OK	OK	OK	OK	OK
32MHz	OK	OK	OK	OK	OK	OK	OK
64MHz	OK	OK	OK	OK	OK	OK	OK

### 【方法2】 旧スタイルのパラメータ設定

```
fx_cor afile [sekibun soffset coffset roffset t1pp smode pp_mode delsizetzoom pmode comment]
```

ここで	<i>afile</i>	–	アプリオリファイル名 0 とすると内部のデフォルトファイル名を使用
	<i>sekibun</i>	–	総積分時間 (sec) 負の値は可能な限りの積分時間 デフォルトまたは 0 はアプリオリファイルに記述している観測時間
	<i>soffset</i>	–	開始時刻のオフセット ( 整数秒単位) デフォルトは 0
	<i>coffset</i>	–	クロックオフセット (sec) ( Y 局が進んでいる場合を正 ) デフォルトは 0.0
	<i>roffset</i>	–	クロックレートオフセット (s/s) デフォルトは 0.0
	<i>t1pp</i>	–	単位積分時間 ( PP ) の設定 (sec) ( デフォルトは 1.0 ) 1 秒以下の値を指定するときは、1 秒を整数で割った値とすること ( 例 : 0.2 )
	<i>smode</i>	–	一度に処理するサンプル数 ( 遅延サーチ範囲 ) モード 0: 最大 200000 点の範囲の相関関数 ( 低速 ) 1: 中 10000 点の範囲サーチ 2: 最小 1000 ( または 2000 ) 点の範囲サーチ ( 高速 ) デフォルトは 2
	<i>pp_mode</i>	–	PP 同期モード 0: PP の開始が秒に同期する 1: 非同期 処理開始可能な時刻から PP を開始 デフォルトは 0
	<i>delsize</i>	–	ラグウインドウサイズ 16, 32, 64, 128, 256, 512, 1024, 2048, .... 0 の場合はプログラム中の DELAYSIZE で示される値

デフォルトは 0 （通常 DELAYSIZE は 32）

*tzoom*    – 時間軸拡大率（整数）  
 0: フルスケール（デフォルト）1 を指定と等価  
 負の値とすると拡大は最大値を中心としたスケールとなる  
 拡大率は絶対値

*pmode*    – プロット表示デバイスモード  
           0 : XWINDOW および PostScript ファイル (pgplot.ps または  
              gnuplot.ps) 出力（デフォルト）  
           1 : PostScript ファイル (pgplot.ps または gnuplot.ps) 出力のみ  
           2 : XWINDOW 出力のみ  
          -1 : グラフ出力無し

*comment* – コメント（グラフ上部に表示）。スペースを含めないこと。  
 省略した場合は会話モード入力になる  
 注：スペースを含むコメントは会話モードで入力すること。

### 【方法 3】 自己相関、予測値ゼロ相関モード

`fx_cor k5file1 [k5file2] [options]`

ここで *k5file1*   – データファイル名  
*k5file2*    – データファイル名  
 ディレクトリ名を省略時は *k5file1* と同じディレクトリにあると見  
 なされる。また *k5file2* を省略時は *k5file2* は *k5file1* と同じとなる。  
*k5file1* と *k5file2* の相互相関処理（*k5file2* を省略時は *k5file1* の自己  
 相関処理となる）が、予測値をゼロとして行われる。RF 周波数も  
 ゼロにセットされる。

*options*   – 方法 1 のオプションが使用可

このモードは以下の予測値ファイルを使用するのと同じである。

```

** This is Apriori file for auto correlation
**

$EXPCODE
APE_ZERO

$STATION1
STATION1 k5file1

$STATION2
STATION2 k5file2

$FREQUENCY
0.0 U
0.0 U
0.0 U
0.0 U

$PCAL_FREQ
0.0
0.0
0.0
0.0

$CLOCK
OFST= 0.0
RATE= 0.0

```

```

$SOURCE
APE_ZERO

$START
00000000000000

$STOP
00000000000000

$APRIORI
PRT=00000000000000
TAU0= 0.0
TAU1= 0.0
TAU2= 0.0
TAU3= 0.0

$END

```

#### 【環境変数のモニター】

fx\_cor で使用する環境変数 K5COUT ( 相関データ出力ディレクトリ )、K5APRIDIR ( 予測値ファイルのディレクトリ )、RGDISP ( PGPLOT のグラフ出力デバイス ) の設定値とプログラムデフォルト値を確認する。

```
fx_cor env
```

#### 4.1.2 実行例: cor ( 1 ビットサンプリングデータ専用 ) による相関処理

予測値ファイル以外はデフォルト値を使用

```

$ cor ./apeXY10.txt      <== カレントディレクトリの予測値ファイル‘apeXY10.txt’を
                           使っての相関処理
                           予測値ファイルのデフォルトディレクトリ
                           (環境変数 K5APRIDIR で設定) 以外を使うときは
                           予測値ファイル名の中にディレクトリを含める
                           相関データはデフォルト出力ディレクトリ
                           (環境変数 K5COUT で設定) に出力される

***** run parameters (Ver 2014.8.13 or later)*****
afile   = ./apeXY10.txt
Total Integ Period (sec) = 0.000000 (0.0 means integrated as scheduled)
T1PP (sec)           = 1.000000   PP_sync mode = 0 (PP sync to 1sec tic)
Lag Window Size      = 32   Search_mode = 2
Start Offset (sec)   = 0
Clock Offset (sec)   = 0.000000e+00   Clock Rate(s/s) = 0.000000e+00
PCAL Detection       : ON
Graphic Out Mode     = 0 (PostScript Out + DISPLAY)   tzoom    = 1
comment = (null)
loop_param = 0 (for regular processing)
ch assign = (1 - 1) (2 - 2) (3 - 3) (4 - 4)
*****
ApeMonit: APRIORI file is NEW VERSION
ApeMonit: EXPCODE   JD0306
ApeMonit: OBS_NUMBER 8
ApeMonit: KASHIM11 /home/kondo/data/testspeed/Xk5data.10.dat
ApeMonit: XYZ -3997505.701700 3276878.404550 3724240.703140
ApeMonit: TOMAKO11 /home/kondo/data/testspeed/Yk5data.10.dat
ApeMonit: XYZ -3680586.301730 2917515.745560 4300987.652680
ApeMonit: BASEID   RH
ApeMonit: PRT      2003 197 2 41 10
ApeMonit: START    2003 197 2 41 5
ApeMonit: STOP     2003 197 2 41 15
ApeMonit: Frequency Table
ApeMonit: Ch1 8209990000.000000 U

```

```

ApeMonit: Ch2 8219990000.000000 U
ApeMonit: Ch3 8249990000.000000 U
ApeMonit: Ch4 8309990000.000000 U
ApeMonit: PCAL Frequency Table
ApeMonit: Ch1 10000.000000
ApeMonit: Ch2 10000.000000
ApeMonit: Ch3 10000.000000
ApeMonit: Ch4 10000.000000
ApeMonit: Frequency Group# 1
ApeMonit: Aprioris
ApeMonit: Tau0 1.540623e-04
ApeMonit: Tau1dot 1.153642e-07
ApeMonit: Tau2dot -1.169617e-12
ApeMonit: Tau3dot -6.131427e-16
ApeMonit: Clock offset 2.485000e-06
ApeMonit: Clock rate 0.000000e+00
ApeMonit: X Clock offset 0.000000e+00
ApeMonit: UT1-UTC (sec) 0.000000
ApeMonit: X-WOBB (asec) 0.000000
ApeMonit: Y-WOBB (asec) 0.000000
ApeMonit: Star 3C273B
ApeMonit: RA 12 29 6.69973194
ApeMonit: DEC 2 3 8.59818480
ApeMonit: EPOCH 2000.0
ApeMonit: GHA 9 46 11.12600000
Directory ../cout already existed!
corr_engine: Version 2011-03-22
corr_engine: maxpp, idsize, irsize, chsize, smode 2048 32 2048 4 2
corr_engine: X Data File is /home/kondo/data/testspeed/Xk5data.10.dat
corr_engine: Y Data File is /home/kondo/data/testspeed/Yk5data.10.dat
checkheader: Header (K5/VSSP) Sync Detected!!
checkheader: File : /home/kondo/data/testspeed/Xk5data.10.dat
checkheader: A/D(bits) 1 CHs 4 SFreq(kHz) 8000 Time 02:41:05 sec 9665
checkheader: Header (K5/VSSP) Sync Detected!!
checkheader: File : /home/kondo/data/testspeed/Yk5data.10.dat
checkheader: A/D(bits) 1 CHs 4 SFreq(kHz) 8000 Time 02:41:05 sec 9665
corr_engine: << runmode (runmode) = 1
corr_engine: << smode (smode) = 2
corr_engine: << # of samples in a unit (usampl) = 4000
corr_engine: << lag size (idsize) = 32
corr_engine: << PP period in sec (t1pp) = 1.0
corr_engine: << # of usampl in a PP (nspp) = 2000
corr_engine: << # of usampl in 1 sec (imax) = 2000
corr_engine: << # of bytes in a usampl (numb) = 2000
corr_engine: << RF offset (Hz) = 0.000000
corr_engine: << Fringe stopping mode = 0 (base band)
xros_engine: << Fringe phase calc mode (modefr) = 3 (3 level approx)
corr_engine: << Fringe phase calc step (frstep) = 32 sample(s)
corr_engine: << Pcal detection mode X station = 1
corr_engine: << Pcal detection mode Y station = 1
corr_engine: << Channel Assignment = (1 - 1) (2 - 2) (3 - 3) (4 - 4)
corr_engine: << Temprary file for header : ./PNRo93hd.tmp
corr_engine: << Temprary file for PP data : ./agxDGypp.tmp
corr_engine: chdif_flag = 0
corr_engine: Atamadashi finished!
corr_engine: temporary info out file (./PNRo93hd.tmp) Opened!
corr_engine: temporary corr out file (./agxDGypp.tmp) Opened!
corr_engine: Start X data time : 9665.000000
corr_engine: Start Y data time : 9665.000000
checkheader: Header (K5/VSSP) Sync Detected!!
checkheader: File : /home/kondo/data/testspeed/Yk5data.10.dat
checkheader: A/D(bits) 1 CHs 4 SFreq(kHz) 8000 Time 02:41:06 sec 9666
corr_engine: PP# 1 data saved
corr_engine: Time elapsed for 1PP processing is 1.029115 sec
corr_engine: X data time (BOPP) : 9665.000000
corr_engine: processed data (1.0/10.0)
checkheader: Header (K5/VSSP) Sync Detected!!
checkheader: File : /home/kondo/data/testspeed/Xk5data.10.dat
checkheader: A/D(bits) 1 CHs 4 SFreq(kHz) 8000 Time 02:41:06 sec 9666
checkheader: Header (K5/VSSP) Sync Detected!!

```

```
checkheader: File : /home/kondo/data/testspeed/Yk5data.10.dat
checkheader: A/D(bits) 1 CHs 4 SFreq(kHz) 8000 Time 02:41:07 sec 9667
corr_engine: PP# 2 data saved
```

..... 中略.....

```
corr_engine: PP# 9 data saved
corr_engine: Time elapsed for 1PP processing is 1.026757 sec
corr_engine: X data time (BOPP) : 9673.000000
corr_engine: processed data (9.0/10.0)
checkheader: Header (K5/VSSP) Sync Detected!!
checkheader: File : /home/kondo/data/testspeed/Xk5data.10.dat
checkheader: A/D(bits) 1 CHs 4 SFreq(kHz) 8000 Time 02:41:14 sec 9674
checkheader: File EOF! (/home/kondo/data/testspeed/Yk5data.10.dat)
corr_engine: Time-elapsed per PP (sec) MIN=1.026757 MAX=1.029326
Postscript out file ==> pgplot.ps <= 作成されるポストスクリプトファイル
```

CH#	FREQ(MHz)	MAX AMP	RESIDUAL DELAY (sec)
1	8209.99	0.001041	-5.69519e-09
2	8219.99	0.000707	-5.19374e-08
3	8249.99	0.000933	4.58736e-09
4	8309.99	0.000835	2.86532e-08

```
COUT File is ../cout/coutt0006.txt <= 作成される相関データファイル
Time elapsed (sec) for One obs process is 13.000000
$
```

処理後、図 4.1 に示すような相関関数が表示される。

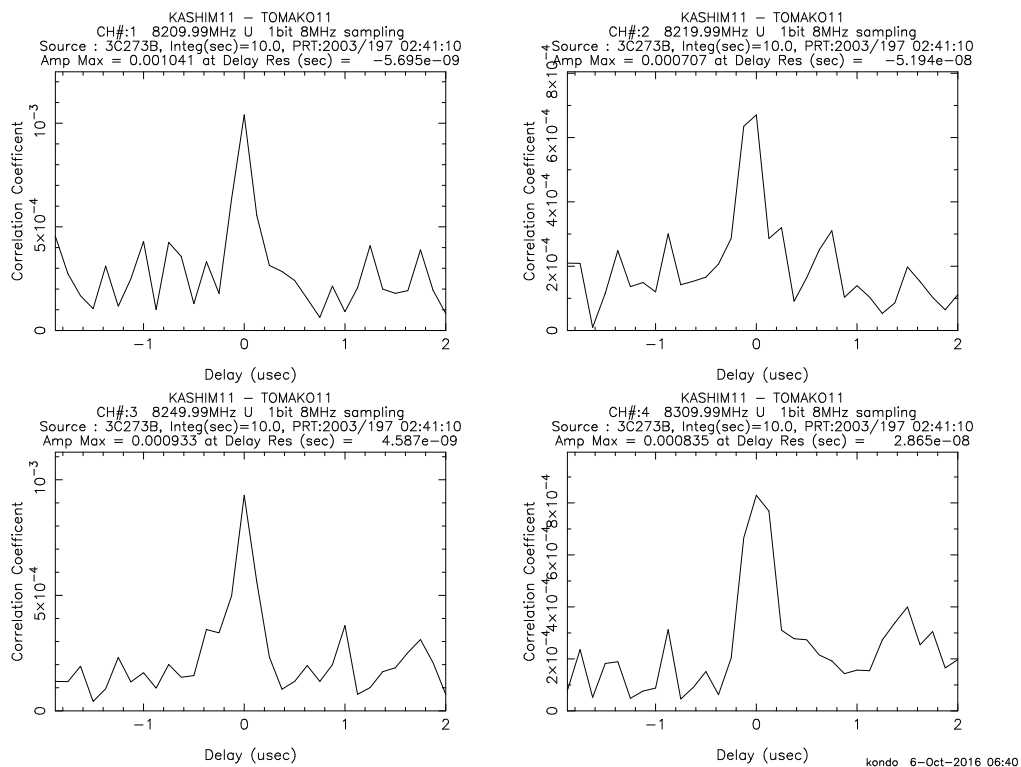


図 4.1: fx\_cor および cor の処理終了後表示される相関関数例

## 4.1.3 実行例: fx\_cor\_new による相関処理 (Mark5B データ同志の処理)

予測値ファイル名以外はデフォルト値を使用

```
$ fx_cor_new ./ape220205919RG.m5b.txt <== Mark5B データ用に作成した予測値ファイルを使う
***** run parameters (Ver 2014.8.13 or later)*****
  afile      = ./ape220205919RG.m5b.txt
  Total Integ Period (sec) = 0.000000 (0.0 means integrated as scheduled)
  T1PP (sec)      = 1.000000   PP_sync mode = 0 (PP sync to 1sec tic)
  Lag Window Size  = 32      Search_mode = 2
  Start Offset (sec) = 0
  Clock Offset (sec) = 0.000000e+00   Clock Rate(s/s) = 0.000000e+00
  PCAL Detection    : ON
  Graphic Out Mode  = 0 (PostScript Out + DISPLAY)   tzoom    = 1
  comment = (null)
  loop_param = 0 (for regular processing)
  ch assign  = (1 - 1) (2 - 2) (3 - 3) (4 - 4) <==
                                     16CH データだがこの部分は4 CH 分しか表示されない

  BPF parameters
    modebpf = 0
  Frequency resolution (MHz) (0 means AUTO) : 0.000000
*****
ApeMonit: APRIORI file is NEW VERSION
ApeMonit: EXPCODE      K10216

... 中略 ...

engine_datafile_open: # of channels to be processed is 16
xros_engine: << runmode (runmode) = 1
xros_engine: << smode (smode) = 2
xros_engine: << # of samples in a unit (usampl) = 1000
xros_engine: << lag size (idsize) = 32
xros_engine: << FFT size for processing = 32
xros_engine: << Lag Window Type = Box
xros_engine: << PP period in sec (t1pp) = 1.0
xros_engine: << # of usampl in a PP (nspp) = 32000
xros_engine: << # of usampl in 1 sec (imax) = 32000
xros_engine: << # of bytes in a usampl (numb) = 2000
xros_engine: << # of bytes in a usampl per CH = 125
xros_engine: << RF offset (Hz) = 0.000000
xros_engine: << Fringe stopping mode = 0 (base band)
xros_engine: << Fringe phase calc mode (modefr) = 9 (9 level approx)
xros_engine: << Fringe phase calc step (firstep) = 8 sample(s)
xros_engine: << Pcal detection mode X station = 1
xros_engine: << Pcal detection mode Y station = 1
xros_engine: << # of channels to be processed = 16
xros_engine: << X data file format = Mark-5B 32MHz 1BIT 16CH <==
                                     データフォーマット情報
xros_engine: << Y data file format = Mark-5B 32MHz 1BIT 16CH
xros_engine: << Channel Assignment = (1 - 1) (2 - 2) (3 - 3) (4 - 4)
                                     (5 - 5) (6 - 6) (7 - 7) (8 - 8)
xros_engine:                                     (9 - 9) (10 - 10) (11 - 11) (12 - 12)
xros_engine:                                     (13 - 13) (14 - 14) (15 - 15) (16 - 16)
xros_engine: << Channel Picked Up = (1 - 1) (2 - 2) (3 - 3) (4 - 4)
                                     (5 - 5) (6 - 6) (7 - 7) (8 - 8)
xros_engine:                                     (9 - 9) (10 - 10) (11 - 11) (12 - 12)
xros_engine:                                     (13 - 13) (14 - 14) (15 - 15) (16 - 16)
xros_engine: << Temprary file for header : ./qRYFBChd.tmp
xros_engine: << Temprary file for PP data : ./q2WioCpp.tmp
xros_engine: << FFT is carried out using FFTW3.0 package
xros_engine: chdif_flag = 0
xros_engine: Atamadashi finished!
xros_engine: temporary info out file (./qRYFBChd.tmp) Opened!
xros_engine: temporary corr out file (./q2WioCpp.tmp) Opened!
xros_engine: Start X data time : 75559.000000
xros_engine: Start Y data time : 75559.000000
xros_engine: PP# 1 data saved

... 中略 ...
```



```

xros_engine: PP# 7 data saved
xros_engine: Time elapsed for 1PP processing is 45.787053 sec
xros_engine: X data time (BOPP) : 75565.000000
xros_engine: processed data (7.0/30.0)
m5b_sync_detect: File EOF! (./R220205919.m5b)
xros_engine: Time-elapsed per PP (sec) MIN=45.747727 MAX=45.791488
Postscript out file ==> pgplot.ps

```

CH#	FREQ(MHz)	MAX AMP	RESIDUAL DELAY (sec)
1	7700.99	0.000834	-4.24985e-11
2	7710.99	0.000935	1.00162e-09
3	7720.99	0.000748	3.14188e-09
4	7850.99	0.000892	1.04695e-09
5	8090.99	0.000854	4.31266e-09
6	8290.99	0.000832	1.02798e-08
7	8490.99	0.000883	7.45903e-09
8	8550.99	0.000833	1.06133e-08
9	8570.99	0.000830	1.49527e-08
10	8580.99	0.000757	1.01071e-08
11	2210.99	0.001171	1.84406e-08
12	2220.99	0.001119	1.64464e-08
13	2240.99	0.001125	1.6231e-08
14	2290.99	0.001223	2.18096e-08
15	2330.99	0.001218	1.84975e-08
16	2340.99	0.001376	1.57303e-08

```

COUT File is ../cout/cout0005.txt
Time elapsed (sec) for One obs process is 333.220641
$

```

処理後、図 4.2 に示すような相関関数 16CH 分が表示される。

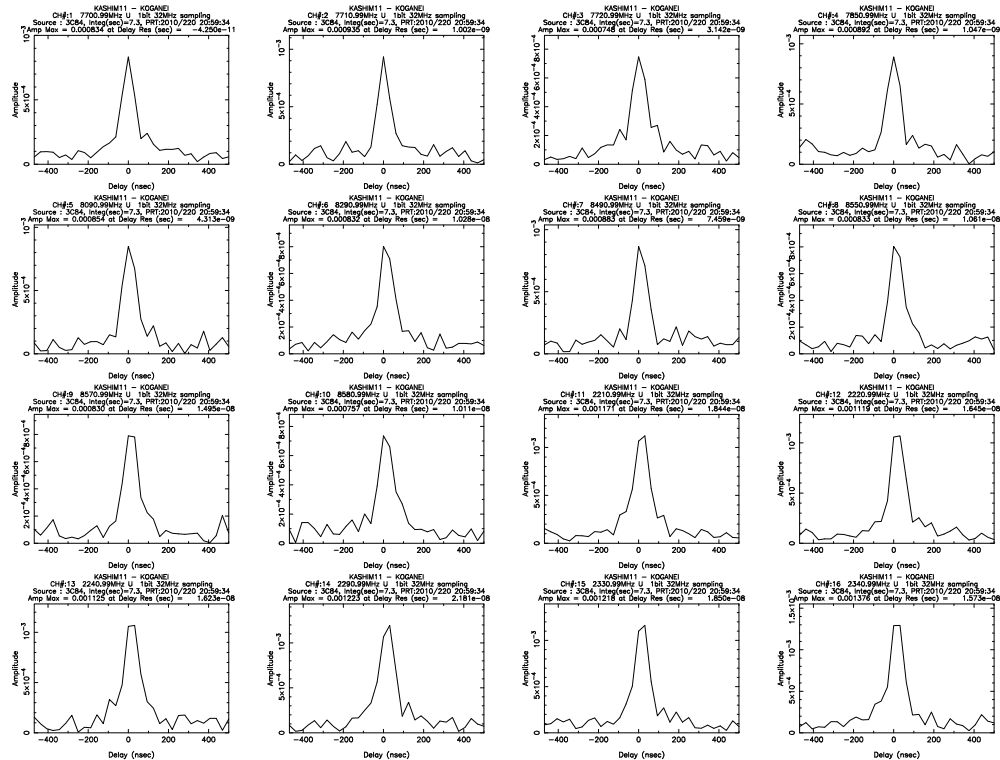


図 4.2: fx\_cor\_new の処理終了後表示される相関関数 (16CH 分) 例

## 4.2 fx\_cor\_all, fx\_cor\_all\_new, cor\_all, cor\_all\_new

### ユーティリティ名

fx\_cor\_all, fx\_cor\_all\_new, cor\_all, cor\_all\_new

### 機能

予測値ファイル名一覧を収納したファイルを使用して複数の予測値ファイルに対する相関処理を実行する。  
予測値ファイル名一覧を収納したファイルの作成法（の一例）は以下の通り。

作成法の一例

予測値ファイルのあるディレクトリで

```
ls -1 ape*.txt > apelist2940U.txt
```

### 4.2.1 実行方法

fx\_cor\_all の場合について説明するがそれ以外のソフトの場合は fx\_cor\_all のところをそのソフトウェア名に置き換える。

fx\_cor\_all *pfile* [*options*]

- |  |  |
|--|--|
| <p>ここで <i>pfile</i></p> <p><i>options</i> (順不同)</p> <p>-integ <i>integration_time</i></p> <p>-coffset <i>clock_offset</i></p> <p>-crate <i>clock_rate</i></p> <p>-soffset <i>start_offset</i></p> <p>-t1pp <i>t1pp</i></p> <p>-pp_nosync</p> <p>-lag <i>delsize</i></p> <p>-pmode <i>pmode</i></p> | <p>– リストファイル名<br/>(中に “apri_calc” で作成された予測値ファイル名を処理順に収納しているもの)<br/>0 とすると内部のデフォルトファイル名を使用</p> <p>– 総積分時間 (sec)<br/>負の値は可能な限りの積分時間<br/>デフォルトまたは 0 はアプリオリファイルに記述している観測時間</p> <p>– クロックオフセット (sec) (Y 局が進んでいる場合を正)<br/>デフォルトは 0.0</p> <p>– クロックレートオフセット (s/s)<br/>デフォルトは 0.0</p> <p>– 開始時刻のオフセット (整数秒単位)<br/>デフォルトは 0</p> <p>– P P 周期 (sec) デフォルトは 1.0<br/>1 秒以下の値を指定するときは、1 秒を整数で割った値とすること<br/>例: 0.2</p> <p>– P P 同期モードを非同期にセット (処理開始可能な時刻から P P を開始)<br/>デフォルトは P P の開始が秒に同期する</p> <p>– ラグウィンドウサイズのセット<br/>例: 16, 32, 64, 128, 256, 512, 1024, 2048, ....<br/>0 の場合はプログラム中の DELAYSIZE で示される値<br/>デフォルトは 0 (通常 DELAYSIZE は 32)</p> <p>– プロット表示デバイスモード<br/>0: XWINDOW および PostScript ファイル (pgplot.ps または gnu-plot.ps) 出力 (デフォルト)</p> |
|--|--|

- 1: PostScript ファイル (pgplot.ps または gnuplot.ps) 出力のみ  
 2: XWINDOW 出力のみ  
 -1: グラフ出力無し
- comment "*any comment*"
- - 
  - コメント (pmode=0,1,2 の場合グラフ上部に表示)
  - nopcal
  - 
  - P C A L 検出機能を抑制する
  - ch1 *ch1Y*
  - 
  - X局の CH1 と相関処理を行うY局の CH を指定 (デフォルトは1)
  - ch2 *ch2Y*
  - 
  - X局の CH2 と相関処理を行うY局の CH を指定 (デフォルトは2)
  - ch3 *ch3Y*
  - 
  - X局の CH3 と相関処理を行うY局の CH を指定 (デフォルトは3)
  - ch4 *ch4Y*
  - 
  - X局の CH4 と相関処理を行うY局の CH を指定 (デフォルトは4)
- 注 fx\_cor\_all\_new および cor\_all\_new の場合は -ch16 *ch16Y*までセット可能
- orule *naming\_rule*
- - 相関出力ファイルの命名則を指定\*
  - 0 : プログラム固定ファイル名 (cout.txt)
  - 1 : coutNNNNtxt (デフォルト)
  - ここで NNNN : 4桁の通し番号
  - 2 : coutEXP\_CODE/coutYYDDDNNNNXYG.txt
  - ここで EXP\_CODE - 実験コード
  - YY - 年 (2桁)
  - DDD - 通日 (3桁)
  - NNNN - スキャン番号 (4桁)
  - XY - 基線 I D (2文字または4文字)
  - G - 周波数グループ (a|b|c|d) または null
  - 3 : coutYYDDDNNNNXYG.txt
  - ここで YY - 年 (2桁)
  - DDD - 通日 (3桁)
  - NNNN - スキャン番号 (4桁)
  - XY - 基線 I D (2文字または4文字)
  - G - 周波数グループ (a|b|c|d) または null
- \* cor\_all では cout の部分が coutt となる
- odir *outdir*
- - 相関出力ファイル作成ディレクトリを指定する
  - デフォルトは環境変数 K5COUT で指定されるディレクトリ、環境変数が指定されていない場合は、../cout ディレクトリ
- frstep *frstep*
- - フリンジ位相計算ステップの設定
  - 0 - 自動設定
  - 1 - 1 サンプル毎
  - 8 - 8 サンプル毎 (デフォルト)
  - N - N サンプル毎 (1000/N は整数であること)
- frauto
- - フリンジ位相計算ステップを自動設定モードに設定 ("frstep 0" と同じ)
- rffset *rf\_offset*
- - X局とY局間の RF 周波数の差を設定 (RFy-RFx)(Hz)
- === 以下は fx\_cor\_all と fx\_cor\_all\_new のみのオプション ===
- modefr *modefr*
- - フリンジストッピング関数近似モードの選択

- 0 - 近似なし
- 9 - 9 レベル近似 ( デフォルト )
- 2 - 2 レベル近似
- 3 - 3 レベル近似
- hanning - ハニング窓関数を使用する ( デフォルトは箱型窓関数 )
- hamming - ハミング窓関数を使用する ( デフォルトは箱型窓関数 )

### 4.3 sdelay

#### ユーティリティ名

sdelay

#### 機能

粗決定サーチを行なう。相関処理データから粗決定サーチを行ない遅延残差および遅延変化率残差を決定する。

#### 4.3.1 実行方法

##### 【方法 1】 ヘルプ機能

sdelay HELP |            使い方が表示される

##### 【方法 2】 会話型のオペレーション

sdelay [PP] [options]

ここで PP – P P 毎の出力を指定するとき。options は方法 4 の項を参照のこと。

##### 【方法 3】 非会話型オペレーション

sdelay *coutfile* [PP] [options]

ここで *coutfile* – 相関処理ファイル。options は方法 4 の項を参照のこと。

##### 【方法 4】 一般的実行方法

sdelay [options]

ここで options ( 順不同可 ) は以下のとおり

- |                       |   |   |
|-----------------------|---|---|
| -v                    | – | 実行途中経過のモニター出力を行う  |
| -cout <i>coufile</i>  | – | 相関処理ファイルのセット  |
| -sdir <i>coutdir</i>  | – | 相関処理ファイルのサーチを開始するディレクトリの指定  |
| -cdir <i>coutdir</i>  | – | 処理すべき相関処理ファイルのあるディレクトリの指定。ここで指定したディレクトリにあるすべての相関処理ファイルが処理される                |
| -ppout                | – | 通常の sdelay 出力ファイルの他に PP 毎の結果を出力   |
| -nosing               | – | 複数の相関処理ファイルを処理したとき、結果をそれぞれのファイル毎に別ファイルで出力する ( デフォルトではすべての結果が一つのファイルに出力される ) |
| -odir <i>outdir</i>   | – | sdelay 出力ファイルのディレクトリを指定   |
| -pgplot <i>device</i> | – | PGPLOT デバイスを指定 ( /NULL を指定するとグラフ出力が抑制される )                                  |
| -ps                   | – | 強制的に PostScript ファイル ( pgplot.ps または gnuplot.ps ) 出力にセット                    |
| -2nd [ <i>t2dot</i> ] | – | 2 次のフリンジ位相変化までサーチする<br>サーチ範囲を $-t2dot \sim +t2dot$ に設定する                    |

- t2dot* 省略時のサーチ範囲は  $\pm 1^{-13} \text{s/s}^2$
- fringe                    -    フリンジ位相と強度の図をプロットする
  - pcal                    -    PCAL 位相と強度の図をプロットする
  - vspeplot                -    ビデオスクロスペクトルを表示する。同時にビデオクロススペクトルデータを vspeout.txt ファイルに出力する
  - no3d                    -    粗決定サーチ関数のプロットを抑制する
  - noplot                  -    PGPLOT デバイスの指定によらず全てのプロットを抑制
  - integ *tinteg*            -    積分時間を強制的に *tinteg* (実数) 秒にセットする
  - vanvleck                -    Van Vleck 補正をラグ毎に行う。使用する式は  $r = \sin(r_c * \pi/2)$  ここで  $r$  は真の相関係数、 $r_c$  は 1 ビットサンプリング後の相関係数。多ビットサンプリングデータに対してはこのオプションは無視される。
  - out *ofile*              -    サーチ関数 (2D データ) をテキストファイルに出力する
  - classic                 -    3D プロットをクラシックスタイル (ティックマークおよび数値なし)
  - tzoom *tzoom*            -    遅延軸のズームファクター ( $tzoom \geq 1.0$ )
  - tshift *tshift*            -    *tzoom* がセットされた際の遅延軸の新たな中央値 (sec) のセット
  - bpf *flow: fhigh[:fact][,flow: fhigh[:fact][,flow: fhigh[:fact][,.....]]]*
    - (fx\_cor と fx\_cor\_new のみ)BPF (バンドパスフィルター) のセット (低域および高域カットオフ周波数による設定: 最大 20 個)
      - flow*    -    ベースバンドでの低域カットオフ周波数 (MHz)
      - fhigh*    -    ベースバンドでの高域カットオフ周波数 (MHz)
      - fact*    -    強度ファクター (0.0–1.0) デフォルトは 1.0
  - bpf2 *fc:bw[:fact][,fc:[bw[:fact][,fc:[bw[:fact][,.....]]]*
    - (fx\_cor と fx\_cor\_new のみ)BPF (バンドパスフィルター) のセット (中心周波数と通過帯域幅による設定: 最大 20 個)
      - fc*        -    ベースバンドでの BPF の中心周波数 (MHz)
      - bw*        -    通過帯域幅 (MHz)。最初の BPF と同じ場合は省略可
      - fact*        -    強度ファクター (0.0–1.0) デフォルトは 1.0
  - fres *fres*                -    BPF 処理時の周波数分解能の設定 (MHz)。デフォルトは自動設定
  - delpcal *ch#:fs[:fi][,ch#:fs[:fi][,.....]]]*
    - 除去する PCAL 信号の周波数をチャンネル毎にセット
      - ch#*        -    チャンネル番号 (1 から開始)
      - fs*        -    PCAL 信号周波数 (MHz)
      - fi*        -    PCAL 信号の間隔 (MHz)
    - 除去される PCAL 信号周波数は  
 $fs, fs + fi, fs + 2fi, fs + 3fi, \dots$   
*fi* を省略した場合は *fs* のみが除去される
    - 除去される帯域幅は FFT の分解周波数幅の 4 倍 (FFT 数は相関処理時のラグ数の 2 倍)。
    - 変更する場合には以下の '-pcalbw' オプションを使用する
- または
- delpcal *delpcal\_file*            -    除去する PCAL 信号情報を記述したファイルを指定する
  - pcalbw *bwmhz*                    -    PCAL 信号除去時に帯域幅をデフォルト値以外に設定 (MHz)

		デフォルト値は FFT 分解周波数幅の 4 倍 (FFT 数は相関処理時のラグ数の 2 倍)
-line[mode]	-	ラインスペクトル処理モードを設定。初段の FFT サイズをラグ数と同じにする。
-hanning	-	相関関数にハニング窓関数を使用する (デフォルトは箱型窓関数)
-lag[size] lag	-	相関処理時とは異なる新たなラグサイズを指定する (例: 32, 64, ...)
		注: '-lag' オプションは '-hanning' オプションや '-vanvleck' オプションと一緒に使えない!
-sub[panels] 1 4 9 16	-	強制的に 1 画面中の副画面数を設定する (PGPLOT の時のみ有効)
-nodel[ay_correction]	-	コヒーレント積分ビデオスペクトル計算時に遅延残差の補正を行わない。デフォルトは補正を行う
-obs[out]	-	観測結果 (予測値 + 残差) とトータル位相の観測値を表示

#### 4.3.2 実行例

```
$ sdelay
SDELAY Ver. 2016-08-12
# of cout files found under ..\cout\ ----- 107
# of cout directories found under ..\cout\ ---- 51
  1 --- Go to File selection
  2 --- Go to further directory selection
Enter your selection --> 2          <==== 2 を選択 (更にディレクトリをサーチ)
Software correlator out directories are as follows
  1 cout
  2 cout021550R
  3 cout021550Rold
  4 cout021550U
  5 cout021550U2
  6 cout021550Uold
  .....
 33 coutCs7200
 34 coutD03C1
 35 coutGIFU
 36 coutGSI
 37 coutJD0306
 38 coutJD0609
  .....
 48 cout_tid06202GY
 49 cout_tid06202GY_8sec
 50 cout_tid062020G
 51 cout_tid062020Y
Select directory --> 36          <===== 36 のディレクトリを選択
Selected directory is ..\cout\coutGSI\
Soft correlator out files are as follows
  1 coutt040970001ACa.txt
  2 coutt040970001ACb.txt
  3 coutt040970001ACc.txt
  4 coutt040970001ACd.txt
  5 coutt040970001TAa.txt
  6 coutt040970001TAb.txt
  .....
 28 coutt042420001TVd.txt
 29 coutt042420002TVa.txt
 30 coutt042420002TVb.txt
 31 coutt042420002TVc.txt
 32 coutt042420002TVd.txt
Select File (0 means all) --> 5    <===== 5 を選択
coutt040970001TAa.txt is selected!
sdelay: output file is ..\sdelayout.txt
sdelay: correlation data file is ..\cout\coutGSI\coutt040970001TAa.txt
fx_cor_out_hd_read: Data File format 7

***** SDELAY (Ver. 2016-08-12) SUMMARY OUT PUT *****
```

```

COUT      : ..\cout\coutGSI\coutt040970001TAa.txt
X DATA   : /vncpc1/ad2/JD0404/T097020000a.dat
Y DATA   : /vncpc1/ad4/JD0404/A097020000a.dat
BASELINE  : TSUKUB32 - AIRA
SOURCE    : 3C454.3          SAMPLING : 1 bit    8 MHz
PRT       : 2004/097 02:00:50   Tinteg(s) : 99.0
LAG SIZE  : 32
CLOCK     : offset -1.085e-006(s) rate -4.542e-014(s/s)
EOP       : utl-utc -0.439965(s)
           : x-wobb -0.140730(asec)
           : y-wobb  0.333260(asec)

```

CH#	FREQUENCY (MHz)	AMP MAX	POSITION ( 64x 128)	RESIDUAL Delay(usec) Rate(ps/s)	SNR
1	8209.99 U	2.757e-003	( 33, 65)	0.000 0.016	77.6
2	8219.99 U	2.822e-003	( 33, 65)	-0.012 0.019	79.4
3	8249.99 U	2.732e-003	( 33, 65)	-0.013 -0.038	76.9
4	8309.99 U	2.627e-003	( 33, 65)	0.014 0.028	73.9

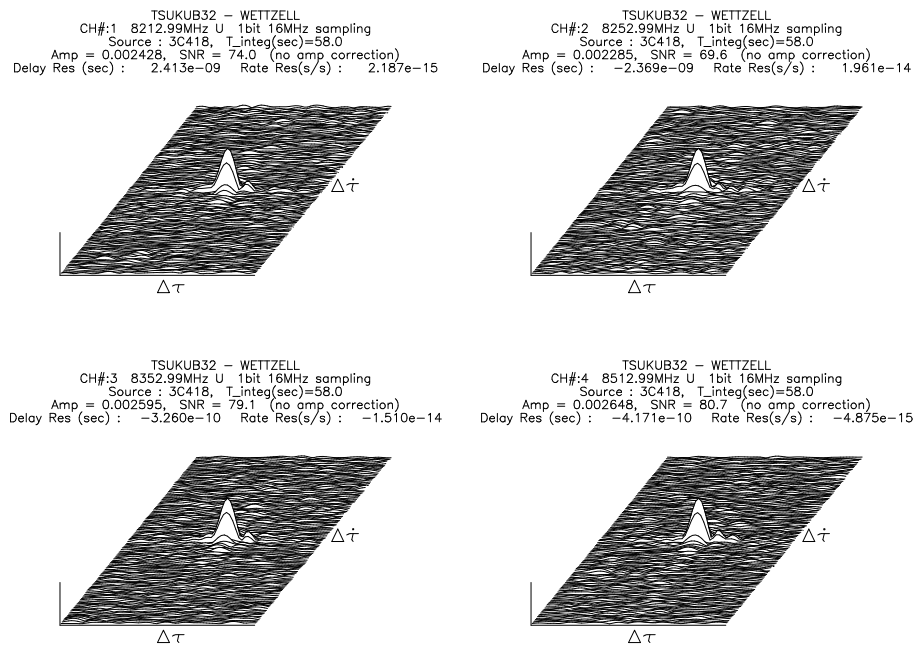
Note: No amplitude correction is made.

CH#	PCAL FREQ(kHz)	X-Amp	X-Phase	Y-Amp	Y-Phase
1	10.00	0.027	132.8	0.062	45.5
2	10.00	0.028	89.4	0.061	-4.1
3	10.00	0.028	-45.4	0.061	-115.5
4	10.00	0.028	161.1	0.060	155.1

\*\*\*\*\*

Outfile is .\sdelayout.txt

処理終了時には図 4.3 の例で示される粗決定サーチ関数が表示される。図 4.4, 4.5, 4.6, には sdelay のオプションで “-fringe” (PP 毎の位相と強度のプロット)、“-pcal” (PCAL 位相と強度のプロット) および “-vspeplot” (ビデオスペクトルプロット) を指定した場合のグラフ出力例 (PGPLOT) を示す。



kondo 14-Nov-2007 17:20

図 4.3: sdelay 処理結果グラフィック出力例



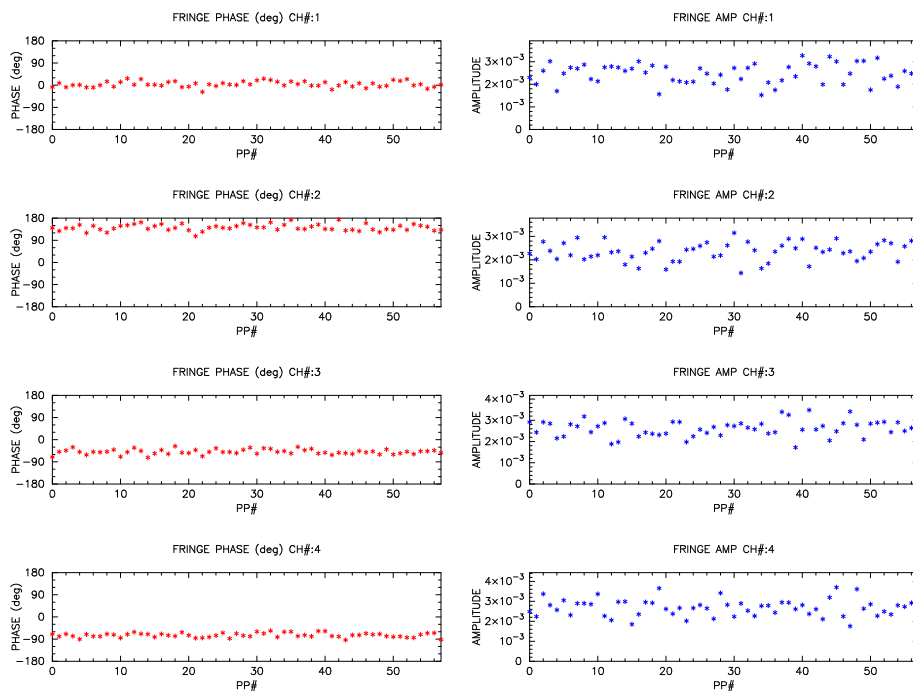


図 4.4: sdelay 処理で “-fringe” (PP 毎の位相と強度のプロット) オプションを指定したときの結果グラフィック出力例

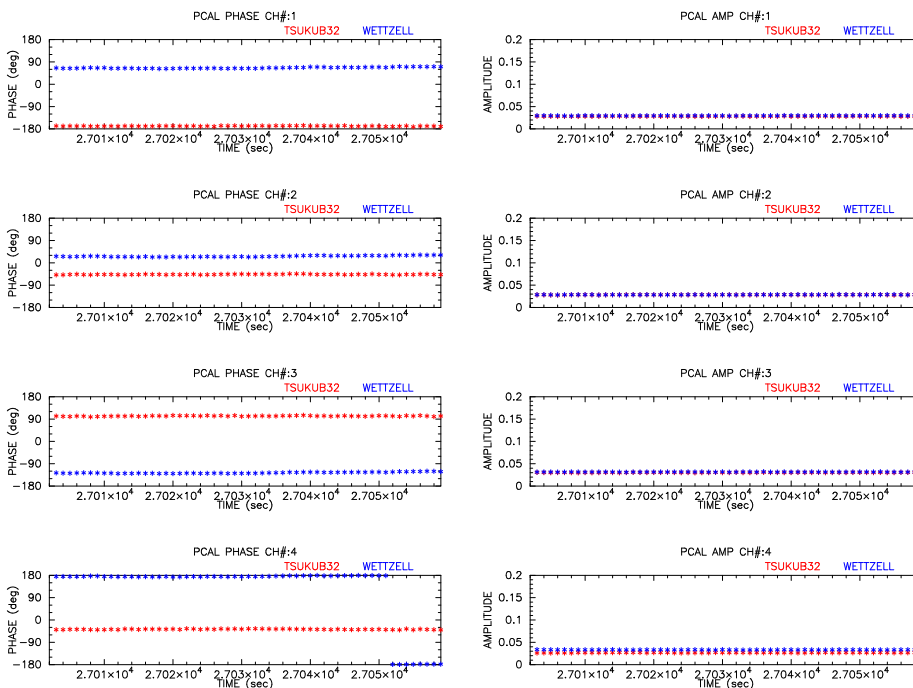


図 4.5: sdelay 処理で “-pcal” (PCAL 位相と強度のプロット) オプションを指定したときの結果グラフィック出力例

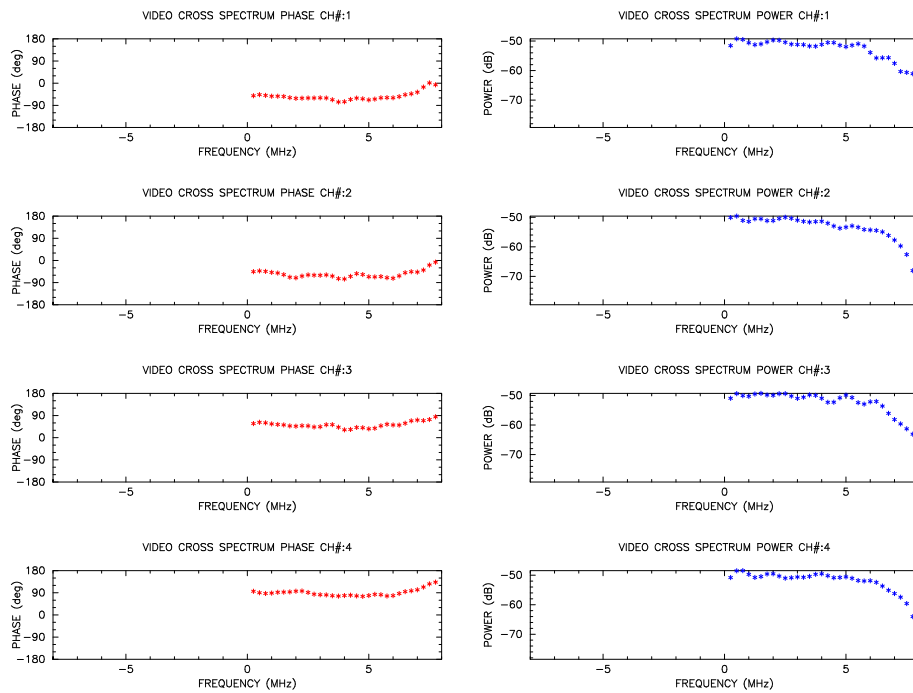


図 4.6: sdelay 処理で “-vspeplot” (ビデオスペクトルプロット) オプションを指定したときの結果グラフィック出力例

“-vspeplot” (ビデオスペクトルプロット) オプションを指定した時の vspeout.txt ファイルに出力されるデータ例。

```
$FORMAT Ver. 2016-12-19
$OBS
$BASELINE HITACHI YAMAGU32
$PRT 2016 300 2 32 30
$PRT (sec in day) 9150.000000
$SOURCE NRA0512C
$SAMPLING(Hz) 1.6e+07
$VIDEO_BW(Hz) 8e+06
$A/D(bits) 2
$NUMBER of PP 300
$SEKIBUN(s) 300.000000
$APRIORI (TAU(s),TAU1dot(s/s),TAU2dot,TAU3dot)
0.00157071683582578 -1.36274927453641e-07 -5.78903382882255e-12 7.24618531073368e-16
$CLOCK (offset(s),rate(s/s))
3.75e-06 0
$FLAG_DELAY_CORRECTION 1
$RESULTS BY CHANNEL
$CHANNEL# 1
$RESULTS freq(MHz), amp, residual_delay(s), err, residual_rate(s/s), err, res_t2dot(s/s^2)
6664.000000 0.00120704 3.07581e-08 8.24096e-10 4.30251e-12 3.2977e-15 0
$VIDEO SPECTRUM INTEGRATED COHERENTLY
$ video freq(Hz) real-part imag-part amp phase(deg)
-8.000e+06 0.000e+00 0.000e+00 0.000e+00 0.000
-7.750e+06 0.000e+00 0.000e+00 0.000e+00 0.000
-7.500e+06 0.000e+00 0.000e+00 0.000e+00 0.000
.....
-5.000e+05 0.000e+00 0.000e+00 0.000e+00 0.000
-2.500e+05 0.000e+00 0.000e+00 0.000e+00 0.000
0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000
2.500e+05 -1.557e-04 5.830e-04 6.034e-04 104.950
5.000e+05 -1.930e-04 1.234e-03 1.249e-03 98.886
.....
7.250e+06 2.905e-04 6.031e-04 6.694e-04 64.285
7.500e+06 1.522e-04 4.461e-04 4.714e-04 71.156
```

7.750e+06    1.468e-05    2.098e-04    2.103e-04    85.997

図 4.7、4.8、4.9、4.10 に GNUPLOT でのグラフ出力例を示す。

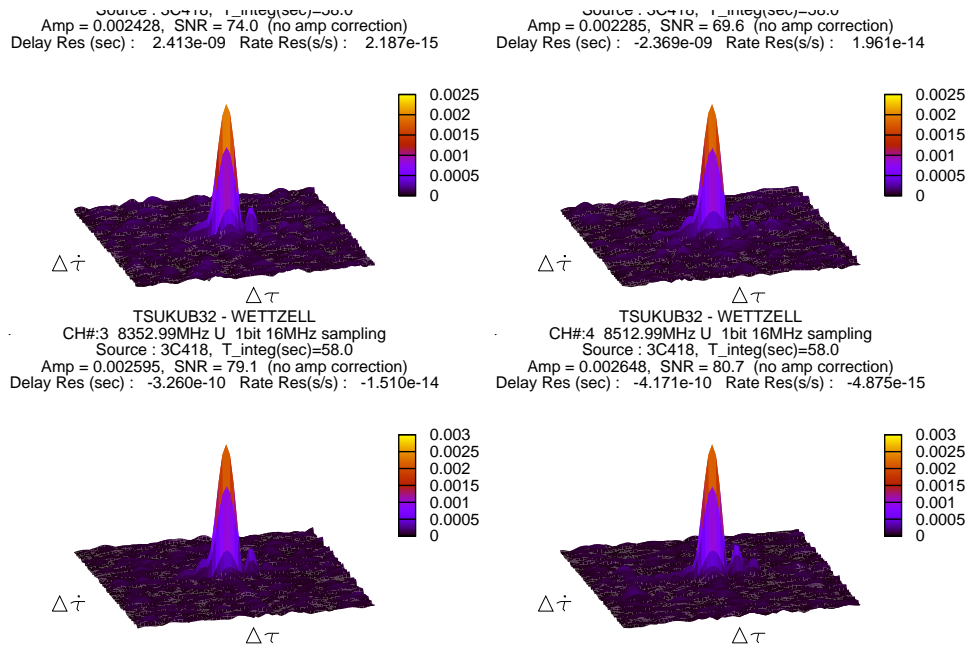


図 4.7: `sdelay` 処理結果グラフィック出力 (GNUPLOT) 例

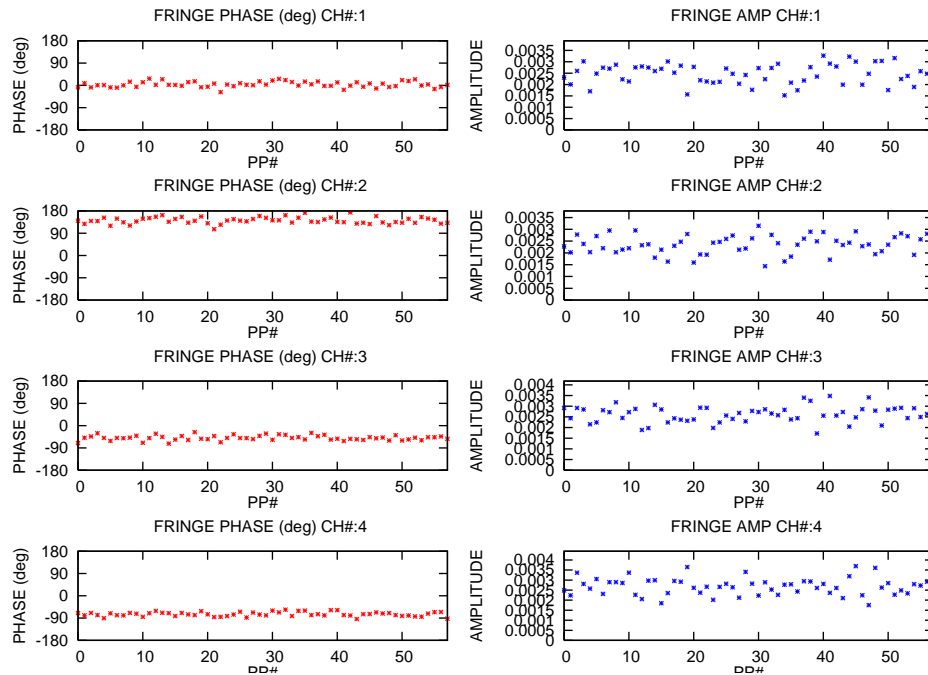


図 4.8: `sdelay` 処理で “-fringe” (PP 毎の位相と強度のプロット) オプションを指定したときの結果グラフィック出力 (GNUPLOT) 例

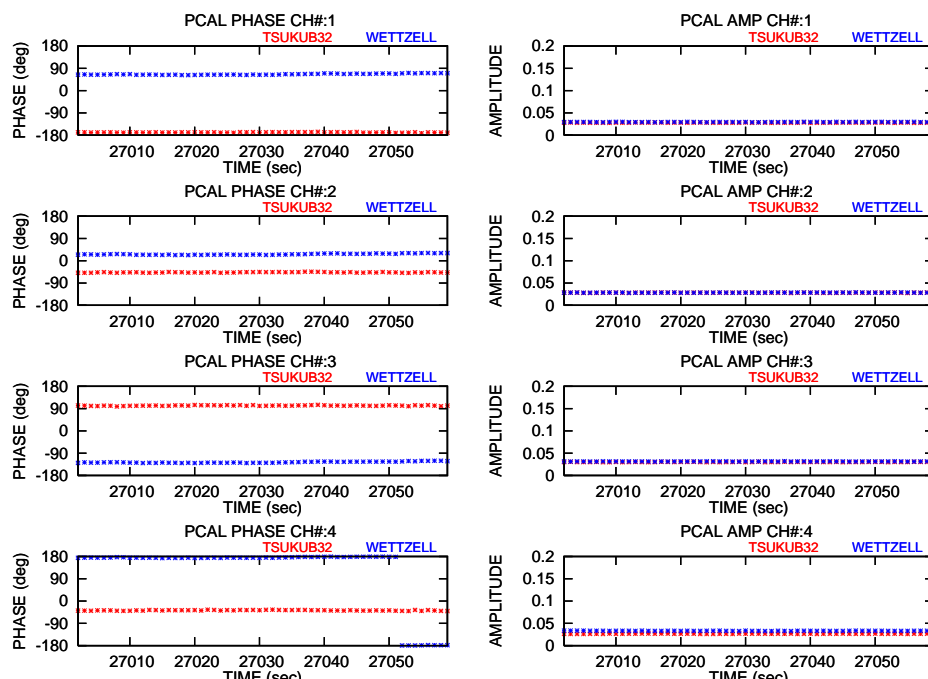


図 4.9: sdelay 処理で “-pcal” (PCAL 位相と強度のプロット) オプションを指定したときの結果グラフィック出力 (GNUPLLOT) 例

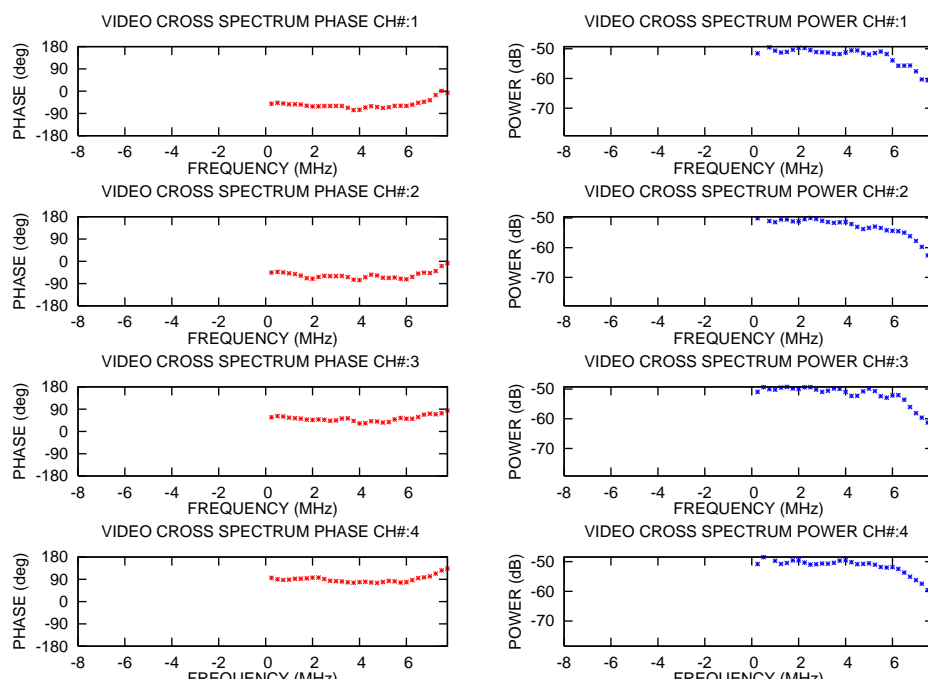


図 4.10: sdelay 処理で “-vspeplot” (ビデオスペクトルプロット) オプションを指定したときの結果グラフィック出力 (GNUPLLOT) 例

## 4.3.3 出力ファイルフォーマット

## 【sdelay デフォルト出力ファイル】

ファイル名 … sdelayout.txt または相関データファイル名の ‘cout’ の部分を ‘sdel’ で置きかえたファイル名

例 相関データファイルが coutt0205.txt の場合、作成されるファイルは sdelt0205.txt

以下ファイルの中身例

```
$OBS
$BASELINE      KASHIM11 TOMAKO11
$PRT           2003 197 2 41 10
$SOURCE        3C273B
$SAMPLING(Hz)  8e+06
$VIDEO_BW(Hz)  4e+06
$A/D(bits)     1
$NUMBER of PP  9
$SEKIBUN(s)    9.000000
$APRIORI (TAU(s) TAU1dot(s/s) TAU2dot TAU3dot)
0.00015406231 1.1536424e-07 -1.1696165e-12 -6.1314267e-16
$CLOCK (offset(s),rate(s/s))
2.485e-06 0
$RESULTS Freq(MHz) AMP Res_Delay(s) Err Res_Rate(s/s) Err Res_2dot(s/s^2) Res_Phase(deg)
8209.99 0.00185916 1.96904e-09 8.73712e-09 4.40734e-13 4.7298e-13 0 65.52
8219.99 0.00139373 -1.03363e-08 1.16548e-08 3.30101e-12 6.3016e-13 0 96.38
8249.99 0.00169796 1.62224e-08 9.56661e-09 5.54919e-13 5.15374e-13 0 114.36
8309.99 0.00188501 -9.87243e-09 8.6173e-09 5.99579e-13 4.6088e-13 0 -114.79
$OBSERVED Freq(MHz) Observed_Delay(s) Observed_Rate(s/s) Total_Phase(deg)
8209.99 0.000154064279043935 1.15364680734089e-07 134.03
8219.99 0.000154051973685111 1.15367541008296e-07 342.32
8249.99 0.00015407853240548 1.15364794919007e-07 240.92
8309.99 0.00015405243756551 1.15364839579305e-07 322.84
```

## 【sdelay PP 毎のデータ出力ファイル】

ファイル名 … 相関データファイル名の ‘cout’ の部分を ‘sdel’ で置きかえたファイル名 + ‘.CH1’ ~ ‘.CH4’

例 相関データファイルが coutt0205.txt の場合、作成されるファイルは sdelt0205.txt.CH1、...、sdelt0205.txt.CH4

以下ファイルの中身例

```
$FORMAT Ver. 2017-02-24
$OBS
$BASELINE      KASHIM11 TOMAKO11
$PRT           2003 197 2 41 10
$PRT (sec in day) 9670.000000
$SOURCE        3C273B
$SAMPLING(Hz)  8e+06
$VIDEO_BW(Hz)  4e+06
$A/D(bits)     1
$NUMBER of PP  9
$SEKIBUN(s)    9.000000
$APRIORI (TAU(s),TAU1dot(s/s),TAU2dot,TAU3dot)
0.00015406231 1.1536424e-07 -1.1696165e-12 -6.1314267e-16
$CLOCK (offset(s),rate(s/s))
2.485e-06 0
$RESULTS freq(MHz), amp, residual_delay(s), err, residual_rate(s/s), err, res_t
2dot(s/s^2)
8209.990000 0.00185916 1.96904e-09 8.73712e-09 4.40734e-13 4.7298e-13 0
$Obtained Tau0,Tau1,Tau2,Tau3
0.000154064279043935 1.153646807e-07 -1.1696165e-12 -6.1314267e-16
$AVERAGED X-PCAL AMP and PHASE(deg), Y-PCAL AMP and PHASE(deg)
0.0688 70.362 0.1595 111.713
$Total PP number and PP period in sec 9 1.000000
$REFERENCE FREQUENCY(MHz) 8209.990000
$ EACH PP DATA
$ Date MOPP_Time tau(sec) amp phs(deg) xpamp xpphs(deg)
ypamp ypphs(deg)
2003/07/16 02:41:05.50 1.5354512615e-04 0.0013034785 85.446 0.0682 71.240
0.1594 111.211
2003/07/16 02:41:06.50 1.5366049550e-04 0.0014538306 88.053 0.0692 73.339
0.1600 111.025
```

```
2003/07/16 02:41:07.50 1.5377586369e-04 0.0020580706 50.928 0.0678 71.786
0.1590 114.300
---以降 PP 数分データが続く
(注: ypamp 以降は実際は 1 行のデータ)
```

### 【sdelay 2D 配列データ出力ファイル】

ファイル名 … ‘-out ofile’ で指定したファイル名 (ofile) 以下ファイルの中身例

```
TSUKUB32 - WETTZELL
CH#:1 2344.99MHz U 1bit 16MHz sampling
Source : 4C39.25, T integ(sec)=54.0
Amp = 0.004531, SNR = 133.2 (no amp correction)
Delay Res (sec) : -2.631e-008 Rate Res(s/s) : 8.603e-013
**** 2D DATA START ****
128 64 < == 配列サイズ 行 (レート方向サイズ) × (ディレイ方向サイズ)
6.12758e-005 4.88404e-005 6.13282e-005 ... < == 1 行目 6 4 点データ
5.81745e-005 6.01026e-005 5.68238e-005 ... < == 2 行目 6 4 点データ
....
3.73513e-005 9.13274e-006 4.24058e-005 ... < == 1 2 8 行目 6 4 点データ
**** 2D DATA END ****

TSUKUB32 - WETTZELL
CH#:2 2352.99MHz U 1bit 16MHz sampling
Source : 4C39.25, T integ(sec)=54.0
Amp = 0.004311, SNR = 126.7 (no amp correction)
Delay Res (sec) : -2.184e-008 Rate Res(s/s) : 1.112e-012
**** 2D DATA START ****
128 64
..... C H 2 データの開始
.....
```

## 4.4 cor\_mon

### ユーティリティ名

cor\_mon

### 機能

相関関数の連続表示を行なう

注: PGPLOT しかサポートしていない (オプション G=PGPLOT で make した場合のみ有効)

### 4.4.1 実行方法

cor\_mon *file\_name* [*options*]

- |     |  |   |
|-----|--|---|
| ここで | <i>file_name</i>                           | 相関データファイル名 (cout 形式または KSP 形式)  |
|     | -z[oom] <i>zoom</i>                        | 遅延方向の拡大係数   |
|     | -a[max] <i>ampmax</i>                      | 縦軸 (強度) の最大値を設定 (デフォルトは 0.001)  |
|     | -d[range] <i>t1 t2</i>                     | 横軸 (遅延) の範囲を <i>t1</i> から <i>t2</i> (sec) に設定する<br>このオプションを設定すると “-z” オプションは無視される |
|     | -h[alt]                                    | 表示を 1 PP 毎に止める (デフォルトは連続表示)   |
|     | -s[msec] <i>sleep_msec</i>                 | 連続モード時の 1 画面毎の休止時間 (msec) を設定する<br>(デフォルトは 200msec)                               |
|     | -ch <i>ch#1[,ch#2[,ch#3[,ch#4[,....]]]</i> | 表示するチャンネルを設定する (例 -ch 1, 4, 3)<br>(デフォルトは全チャンネル)                                  |

## 4.4.2 実行例

図 4.11 に 16CH の相関データを表示した例を示す。

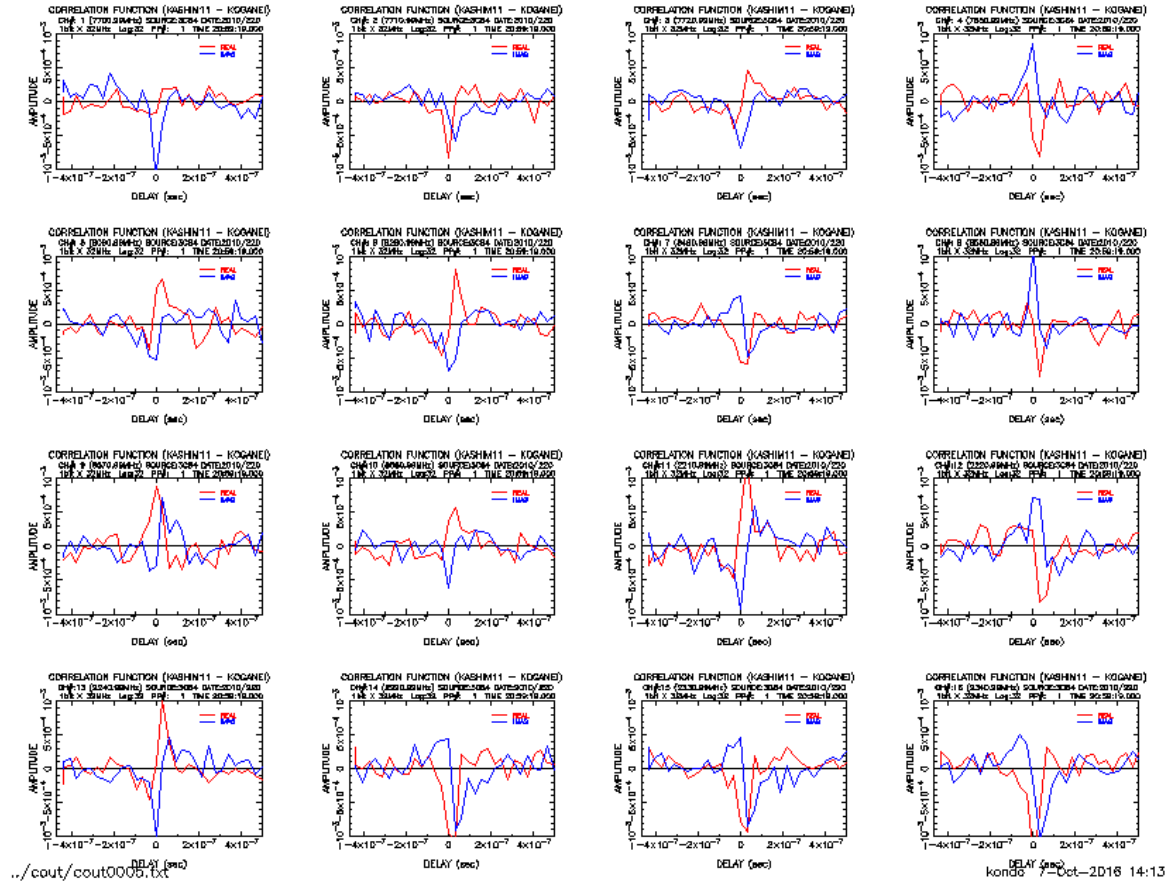


図 4.11: cor\_mon で 16CH 相関データを表示した例。赤線は実部、青線は虚部を示す。

## 5 データチェック関係ソフトの使い方

### 5.1 oscillo

ユーティリティ名

oscillo

機能

サンプリングデータ時系列の連続表示を行なう (K5/VSSP 用, PGPLOT 専用)

#### 5.1.1 実行方法

oscillo *file\_name* [*options*]

- |     |                            |  |
|-----|----------------------------|--|
| ここで | <i>file_name</i>           | サンプリングデータファイル名 (K5/VSSP)                   |
|     | -t[span] <i>tspan</i>      | 時間軸幅 (sec)                                 |
|     | -h[alt]                    | 表示を 1 掃引毎に止める (デフォルトは連続表示)                 |
|     | -s[msec] <i>sleep_msec</i> | 連続モード時の 1 画面毎の休止時間 (msec) を設定する (デフォルトは 0) |

#### 5.1.2 実行例

図 5.1 に 64MHz×2bit×4ch データを表示した例を示す。

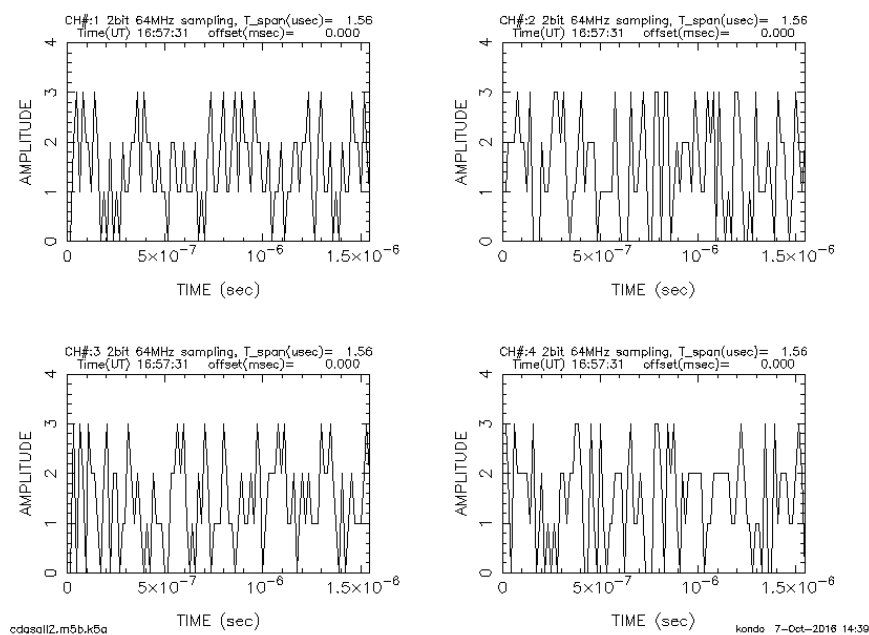


図 5.1: oscillo で 64MHz×2bit×4ch データを表示した例。



## 5.2 speana

ユーティリティ名

speana

機能

サンプリングデータのスペクトル表示を行なう (K5/VSSP 用)

### 5.2.1 実行方法

speana *data\_file* [*options*]

- |  |  |
|--|--|
| <p>ここで <i>data_file</i><br/><i>options</i> (順不同)<br/>-m[ode] <i>mode</i></p> | <p>– サンプリングデータファイル名 (K5/VSSP フォーマット)</p> <p>– 座標軸スケールモード<br/><i>mode</i> : YX<br/>X: 横 (周波数) 軸モード<br/>=0: ログスケール<br/>=1: リニアスケール (デフォルト)<br/>Y: 縦 (強度) 軸モード<br/>=0: ログスケール (デフォルト)<br/>=1: リニアスケール (すべての CH で共通)<br/>=2: リニアスケール (CH 毎独立)<br/>–value: 自己相関 (中心の <i>value</i> 点表示)<br/>–1: 自己相関 (フルレンジ)</p> |
| <p>-pmode <i>pmode</i></p>   | <p>– プロット表示デバイスモード<br/>0: XWINDOW および PostScript ファイル (pgplot.ps または gnuplot.ps) 出力 (デフォルト)<br/>1: PostScript ファイル (pgplot.ps または gnuplot.ps) 出力のみ<br/>2: XWINDOW 出力のみ</p>   |
| <p>-se[kibun] <i>sekibun</i></p>   | <p>– 総積分時間 (sec)</p>   |
| <p>-ti[ntegration] <i>sekibun</i></p>  | <p>– 総積分時間 (sec)</p>   |
| <p>-i[ntegration] <i>sekibun</i></p>   | <p>– 総積分時間 (sec)</p>   |
| <p>-co[mment] <i>comment</i></p>   | <p>– コメント (グラフ上部に表示)</p>   |
| <p>-soffset <i>soffset</i></p>   | <p>– 開始時刻のオフセット (デフォルトは 0.0)</p>   |
| <p>-toffset <i>soffset</i></p>   | <p>– 開始時刻のオフセット (デフォルトは 0.0)</p>   |
| <p>-offset <i>soffset</i></p>  | <p>– 開始時刻のオフセット (デフォルトは 0.0)</p>   |
| <p>-f1[khz] <i>f1khz</i></p>   | <p>– プロット周波数範囲の下限周波数 (kHz) (デフォルトは 0.0)</p>  |
| <p>-f2[khz] <i>f2khz</i></p>   | <p>– プロット周波数範囲の上限周波数 (kHz) (デフォルトは最大ビデオ周波数)</p>  |
| <p>-min[dbm] <i>mindbm</i></p>   | <p>– 表示強度の最小値 (dBm) (デフォルトは自動)</p>   |
| <p>-max[dbm] <i>maxdbm</i></p>   | <p>– 表示強度の最大値 (dBm) (デフォルトは自動)</p>   |
| <p>-nops</p>   | <p>– postscript 出力の抑制。 “-p 2” と同じ</p>  |

### 5.2.2 実行例

図 5.2 に 64MHz×2bit×4ch データのスペクトルを表示した例を示す。

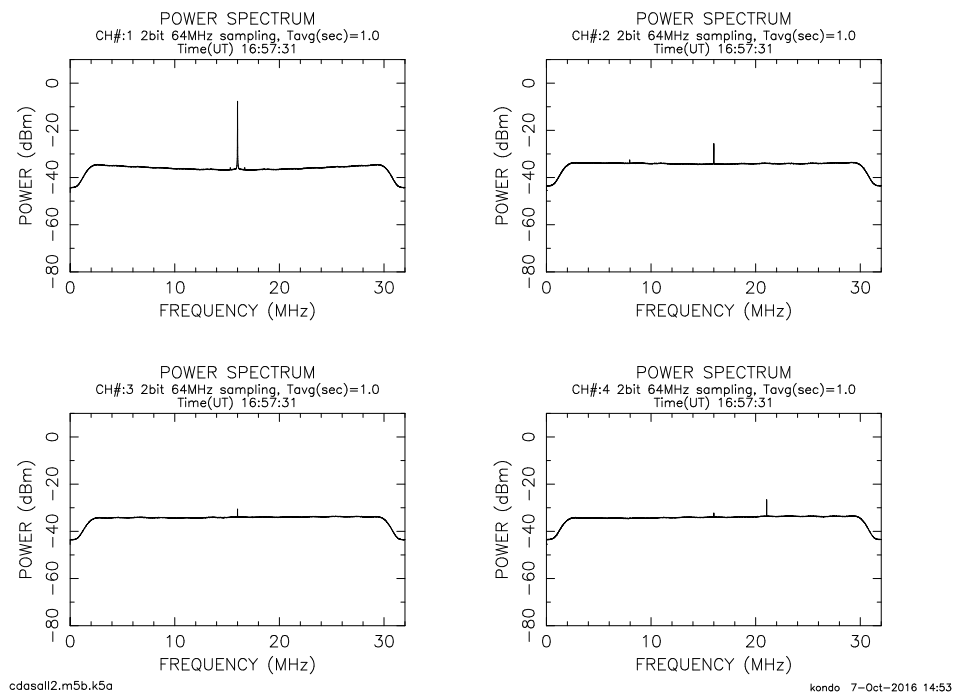


図 5.2: speana で 64MHz×2bit×4ch データのスペクトルを表示した例。

## 5.3 g\_speana

### ユーティリティ名

g\_speana

### 機能

各種フォーマットのサンプリングデータのスペクトル表示を行なう。サンプリングデータフォーマットは K5/VSSP 以外に Mrak-5B, VDIF、ADS が可能。

#### 5.3.1 実行方法

g\_speana *data\_file* [*options*]

- |     |                      |   |                           |
|-----|----------------------|---|---------------------------|
| ここで | <i>data_file</i>     | – | サンプリングデータファイル名 (各種フォーマット) |
|     | <i>options</i> (順不同) |   |                           |
|     | -m[ode] <i>mode</i>  | – | 座標軸スケールモード                |

```

mode : YX
      X: 横 (周波数) 軸モード
          =0: ログスケール
          =1: リニアスケール (デフォルト)
      Y: 縦 (強度) 軸モード
          =0: ログスケール (デフォルト)
          =1: リニアスケール (すべての CH で共通)
          =2: リニアスケール (CH 毎独立)
      -value: 自己相関 (中心の value 点表示)
      -1: 自己相関 (フルレンジ)

-pmode pmode      - プロット表示デバイスモード
                   0: XWINDOW および PostScript ファイル (pgplot.ps または gnu-
                   plot.ps) 出力 (デフォルト)
                   1: PostScript ファイル (pgplot.ps または gnuplot.ps) 出力のみ
                   2: XWINDOW 出力のみ

-se[kibun] sekibun      - 総積分時間 (sec)
-ti[ntegration] sekibun - 総積分時間 (sec)
-i[ntegration] sekibun - 総積分時間 (sec)
-co[mment] comment      - コメント (グラフ上部に表示)
-so[ffset] soffset      - 開始時刻のオフセット (デフォルトは 0.0)
-to[ffset] soffset      - 開始時刻のオフセット (デフォルトは 0.0)
-o[ffset] soffset      - 開始時刻のオフセット (デフォルトは 0.0)
-f1[khz] f1khz          - プロット周波数範囲の下限周波数 (kHz) (デフォルトは 0.0)
-f2[khz] f2khz          - プロット周波数範囲の上限周波数 (kHz) (デフォルトは最大ビデオ
                        周波数)

-min[dbm] mindbm        - 表示強度の最小値 (dBm) (デフォルトは自動)
-max[dbm] maxdbm        - 表示強度の最大値 (dBm) (デフォルトは自動)
-nops                   - postscript 出力の抑制。 “-p 2” と同じ
-1ch ch#                - 1CH モードに強制的に設定し CH#をセットする
-4ch ch#1 ch#2 ch#3 ch#4
                        - 4CH モードに強制的に設定し CH#をそれぞれセットする
-ch ch#1 ch#2 .....    - 任意の CH をピックアップする
-m5b | -vdif | -ads | -oct
                        - サンプリングデータフォーマットを指定する (デフォルトは
                        K5/VSSP フォーマット:VSSP32、VSSP64 を含む)

==(以下は ADS フォーマットまたは M5B フォーマットの場合)==
-numch numch            - データファイルの CH 数の設定
-adbit adbit            - AD ビット数の設定
-sf[req] sfMHz          - サンプリング周波数 (MHz) の設定
-st[ime] YYYYDDDHHMMSS | YYYY/DDD-HH:MM:SS
                        - データ開始日付および時刻の設定 (ADS フォーマットの場合)。デ
                        フォルトは ADS ファイル名から取得

```

## 5.3.2 実行例

Mark5B データのスペクトル表示例。

```
$ g_speana cdasall2.m5b -m5b <== Mark5B データであることを明示するため '-m5b' オプションを使う。VDIF データの場合は '-vdif' を使う
g_speana Ver. 2016-06-17 compiled with FFTW3.0
compiled for PGLOT

Data File is cdasall2.m5b
File cdasall2.m5b opened (638349270 bytes)
Time(UT) MJD 226 16:57:31
  fname      : cdasall2.m5b
  data file start time      : Time(UT) MJD 226 16:57:31
  data pick-up start time   : Time(UT) MJD 226 16:57:31
  sfkHHz      : 64000
  adbit       : 2
  numch       : 16
  plot numch   : 16
  plot ch#     : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
  sekibun     : 1.000000
  soffset     : 0.000000
CH# 1: Maximum data (dB, dBm) is -16.716162 -7.716162
CH# 2: Maximum data (dB, dBm) is -34.588219 -25.588217
CH# 3: Maximum data (dB, dBm) is -39.548599 -30.548597
CH# 4: Maximum data (dB, dBm) is -35.545822 -26.545824
CH# 5: Maximum data (dB, dBm) is -41.150509 -32.150509
CH# 6: Maximum data (dB, dBm) is -41.239742 -32.239742
CH# 7: Maximum data (dB, dBm) is -42.367371 -33.367371
CH# 8: Maximum data (dB, dBm) is -36.048561 -27.048559
CH# 9: Maximum data (dB, dBm) is -36.417774 -27.417774
CH# 10: Maximum data (dB, dBm) is -41.161625 -32.161625
CH# 11: Maximum data (dB, dBm) is -42.641129 -33.641129
CH# 12: Maximum data (dB, dBm) is -36.593994 -27.593994
CH# 13: Maximum data (dB, dBm) is -40.218788 -31.218788
CH# 14: Maximum data (dB, dBm) is -36.009422 -27.009422
CH# 15: Maximum data (dB, dBm) is -42.393009 -33.393009
CH# 16: Maximum data (dB, dBm) is -37.771706 -28.771704
All CH: Maximum and Minimum data (dBm) are -7.716162 -49.829594
Time elapsed for processing is 50.265196 sec
Created PostScript file ==> pgplot.ps
  Type <RETURN> for next page: <== まずは 4ch 分が表示されるので、リターンキーを押すたびに更に 4ch 分ずつ表示される
$
```

図 5.3 に Mark5B データのスペクトル (最後の 4ch 分) を表示した例を示す。

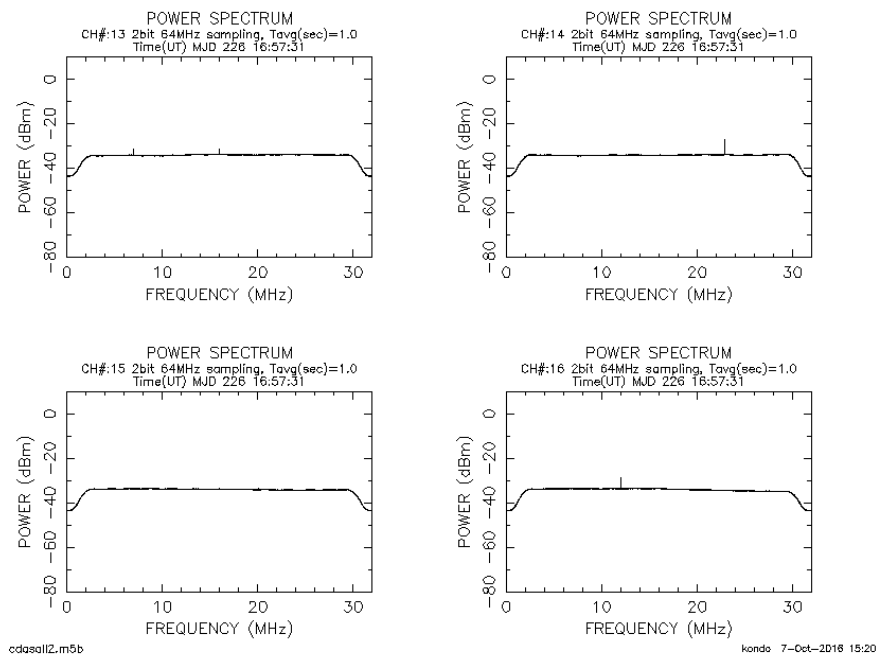


図 5.3: g\_speana で Mark5B データのスペクトル (最後の 4ch 分) を表示した例。

## 5.4 datachk

### ユーティリティ名

datachk

### 機能

ヘッダー部を頼りにヘッダー間のデータビット数をカウントすることにより、ビットスリップまたはビットメイクがあったかどうかでデータをチェックする。またアナログ信号が+のサインであった割合を%で表示することも可能である。更にエラーが起こったデータファイルの記録機能も有する。この機能は、サンプラーの連続試験時にエラーが起こったデータファイルだけ保存したい場合に便利な機能である。

なお、このチェックで AUX MISALIGN (misaligned AUX field) と診断された K5/VSSP32 データは aux\_recov を使って修復が可能である。

データの中身については例えば od コマンド等で確認する。

### 5.4.1 実行方法

```
datachk file_name [mode [logfile [errlog [keepmode]]]]
```

ここで *file\_name*    - K5/VSSP データファイル名 (デフォルトは tds.data)  
           *mode*       - サンプリング統計 (ゼロバランス) 表示モード

- 0 : サンプル統計は表示しない(デフォルト)  
最初と最後のフレーム情報およびエラーの生じたフレーム情報のみを表示
- 1 : サンプル統計を表示する
- 2 : モード0と同じ。ただし全フレーム表示する
- 3 : AD レベルごとの度数分布を1秒ごとに表示
- 4 : AD レベルごとの度数分布を表示(全データの結果)
- logfile* - モード0の場合にチェック結果のサマリーを出力するファイル名。  
このファイル名がの先頭が“-”の場合サマリー出力は既存のファイルに追加されていく。  
デフォルトはサマリー出力なし。
- errlog* - モード0の場合にエラーが発生したデータファイルの情報を出力するファイル名。  
このファイル名がの先頭が“-”の場合、出力は既存のファイルに追加されていく。  
チェックしたデータファイルにエラーが無い場合は、このファイルは作成(出力)されない。デフォルトはエラーログ出力なし。
- keepmode* - モード0の場合にエラーの生じたデータファイルの保存モード
  - 0 : 何もしない(デフォルト)
  - 1 : データファイルの名前を元の名前 + “.NNNN.err” に変更する。
  - 2 : データファイルを 元の名前 + “.NNNN.err” にコピーする。  
ここでNNNNは0001から9999で繰り返す  
この通し番号はdatachkを実行するディレクトリ下の  
“counter\_file\_datachk.tmp” という名のテキストファイルで管理する。

ゼロバランス表示はアナログ信号が+のサインであった割合を%で表示

#### 5.4.2 サマリーファイル

datachk をオプション *logfile* を指定して実行すると サマリーファイルが出力される。そのファイル例を以下に示す

```
# File Name:
D:\IPVLBI\data\test02.dat
# FMT  A/D CH  f(kHz)  LPF(MHz):
VS32  1    1    32000      16
# Start and Last Time:
2006/318 23:20:28 84028
2006/318 23:25:27 84327
# Duration:
300
# Byte offset of 1st header:
0
# STATISTICS total bad discon discon_with_bitslip aux_sep EFLG:
300 1 0 0 147 0
# BIT SLIP:
1 26432
```

### 5.4.3 エラーログファイル

datachk をオプション *errlog* を設定して実行するとエラーログファイルが出力される。そのファイル例を以下に示す

keepmode=0 の場合

```
# Errored Data File Name:
test02.dat
# FMT  A/D CH  f(kHz)  LPF(MHz):
VS32  1    1   32000    16
```

keepmode=1 の場合 (リネームモード)

```
# Errored Data File Name:
test02.dat
# FMT  A/D CH  f(kHz)  LPF(MHz):
VS32  1    1   32000    16
# Renamed to:
test02.dat.0006.err
```

keepmode=2 の場合 (コピーモード)

```
# Errored Data File Name:
test02.dat
# FMT  A/D CH  f(kHz)  LPF(MHz):
VS32  1    1   32000    16
# Copied to:
test02.dat.0007.err
```

## 5.5 vdifcheck

ユーティリティ名

vdifcheck

機能

VDIF フォーマットデータの中身のチェックを行う。

### 5.5.1 実行方法

vdifcheck *vdif\_file* [*options*]

- |   |  |
|---|--|
| <p>ここで <i>vdif_file</i><br/> <i>options</i> (順不同)<br/>         -f[ormat] 0 1 2 3 99<br/> <br/>         -d[mode] 0 1 2</p> | <ul style="list-style-type: none"> <li>- サンプルングデータファイル名 (VDIF フォーマット)</li> <li>- VDIF フォーマットの設定           <ul style="list-style-type: none"> <li>0 : 自動設定 (デフォルト)</li> <li>1 : レガシーモード (拡張データは表示しない)</li> <li>2 : 自動設定。ただし edv# は考慮しない</li> <li>3 : edv# を強制的に Kashima フォーマットに設定する</li> <li>99 : デバッグ用</li> </ul> </li> <li>- 表示モードの設定           <ul style="list-style-type: none"> <li>0 : 最初と最後のフレームのみ表示 (デフォルト)</li> <li>1 : すべてのフレームを表示</li> <li>2 : 最初と最後のフレームおよび毎正秒のフレームを表示</li> </ul> </li> </ul> |
|---|--|

- n[frame] *nmax*            - 読み込むフレーム数を設定 (デフォルトはすべて)
- l[legacy]                - ヘッダーバイト数をレガシーモード (16 バイト) に強制的に設定
- s[kip] *n*                - 最初の *n* バイトを読み飛ばす
- k[ashima] [*bytesframe*] - ヘッダーフォーマットを Kashima フォーマットに強制的に設定する  
                               *bytesframe*: フレーム長 (バイト) を強制的に設定
- pup *thread\_num*        - スレッド#を指定する (デフォルトは最初に出現したスレッド#)
- h[histogram]            - AD レベル毎の出現頻度を表示する

### 5.5.2 実行例

```
$ vdifcheck ./test_sv.vdif -h
*****
*      VDIF data check      *
*      Ver 1.20  2019-10-12  by  T.KONDO  *
*      *      *      *      *
*****

Data File : ./test_sv.vdif
now analyzing # of threads .....
# of threads included is 1
thread ID   : 1
# of frames in a second : 16000
Thread ID to pick up ==> 1
1st header information is as follows

===== VDIF HEADER INFORMATION =====
VDIF header (raw)      : 0012F2DD 2700088F 230003EC 04010062
extended header (raw) : ABADDEED 00000000 00000000 00000000

Invalid flag = 0 :      Legacy flag = 0
Sec from ref epoch = 1241821 : Ref epoch = 39  (2019/07/15 08:57:01)
Frame # = 2191 : VDIF Ver# = 1 : # Chs(log2) = 3
Frame length in 8 byte unit = 1004 (= 8032 bytes)
Data type = 0 : #bits/sample-1 = 1 : Thread ID = 1
Station ID (A2) = b      (= 98 in number)
EDV : -85
Exteded Data W1 W2 W3 W4 : ADDEED 00000000 00000000 00000000
--- in case of NICT Extended Format ---
uflag = 0 :  srate = 0 :  Sync block = 00000000
DAS/station name (A8) :

=====

===== EXTENDED DATA =====
HH:MM:SS  FRAME#  I  L  VER  CHS      #Bytes  C AD  TID SID  EXTENDED DATA
                                     W1      W2
-----
08:57:01   2191   0  0   1    8        8032  0  2    1  b 00ADDEED 00000000
.....
08:57:09   1681   0  0   1    8        8032  0  2    1  b 00ADDEED 00000000
=====

===== PROBABILITY(%) OF OCCURRENCES =====
LEVEL  CH01  CH02  CH03  CH04  CH05  CH06  CH07  CH08
-----
0       9.3  11.2  25.9  28.0  28.4  28.6  28.4  28.9
1      40.7  38.8  24.1  22.0  21.6  21.4  21.6  21.1
2      40.7  38.8  24.1  22.0  21.6  21.4  21.6  21.1
3       9.3  11.2  25.9  28.0  28.4  28.6  28.4  28.9
=====

===== SUMMARY =====
File Name   : ./test_sv.vdif
Size        : 1024000000 bytes
# of threads: 1
Start Time  : 2019/07/15 08:57:01
```



```

Thread ID : 1      Frame# : 2191
Bytes/Frame : 8032 bytes
Data type   : Real Data      AD bits : 2    #Channels : 8
EDV#        : -85
# of frames/sec : 16000
# of total frames : 127491  (#bad frames : 0)
Data period (sec)/thread : 7.968187
Data Rate/thread (Mbps) : 1024.000000
Sampling Frequency Estimated (MHz) for real : 64.000000
=====

$

```

## 5.6 m5check

### ユーティリティ名

m5check

### 機能

Mark5 フォーマットデータの種類および中身のチェックを行う。

#### 5.6.1 実行方法

m5check *m5file* [*mode*]

または

m5check *m5file* [*options*]

<p>ここで <i>m5file</i></p> <p><i>mode</i></p> <p><i>options</i> (順不同)</p> <p>-a[ll]</p> <p>-old</p> <p>-f</p> <p>-t[rack] <i>ntrack</i></p> <p>-vlba</p> <p>-p[arity]</p> <p>-nrzm</p> <p>-s[kip] <i>samples</i></p>	<p>– サンプリングデータファイル名 (Mark5 フォーマット)</p> <p>– チェックするフォーマットの設定</p> <p>0 : パリティ有 8 16 32 64 ビットワードモード</p> <p>1 : パリティ無し 8 16 32 64 ビットワードモード</p> <p>2 : 古いバージョンスタイルの表示</p> <p><i>mode</i> を省略するとすべてのモードを自動チェック</p> <p>– すべてのデータをチェック (Mark5B フォーマットの場合のみ)</p> <p>– 古いバージョンの スタイルでの実行</p> <p>– シンクブロックのチェックを省略。もし以下のオプションのいずれかを設定した場合はこのオプションは無視される</p> <p>– トラック数を設定 (8 16 32 64)。デフォルトは 32</p> <p>– VLBA フォーマットモードに設定 (デフォルトは mark 5)</p> <p>– パリティ有りモードに設定 (デフォルトはパリティ無し)</p> <p>– NRZM モードに設定 (デフォルトは non NRZM モード)</p> <p>– 最初に読み飛ばすサンプル数を設定</p>
--	--

#### 5.6.2 実行例

Mark5B をチェックした例

```
kondo@io:~/chkdata/m5b$ m5check JPddcall.m5b
```

```

*****
*      Mark-5 data structure analysis      *
*      Ver 1.82  2012-11-26  by  T.KONDO/NICT      *
*      *      *      *      *      *      *      *
*****

1st STEP: Checking Mark5B format
Data File : JPddcall.m5b
Now analyzing the data file is Mark5B or not .....

OK this is Mark 5B Format data

          TIME CODE  CRC-
FRAME#  SYNC  UUUUTFFF MJDsssss.ssss 16  DATA#1  DATA#2  Mbps
00000  ABADDEED  F00F0000 22661050 0000A474 5CB997F7 538A10F3  N.A.
00001  ABADDEED  F00F0001 22661050 0000A474 5B5B6BB7 994A5166  N.A.
00002  ABADDEED  F00F0002 22661050 00012471 8975153B 055BC52A  N.A.
00003  ABADDEED  F00F0003 22661050 00012471 7DAD8195 1744AD2E  N.A.
00004  ABADDEED  F00F0004 22661050 0002247B E0E5EE59 19526019  N.A.
00005  ABADDEED  F00F0005 22661050 0002247B 185A2A95 82166A5D  N.A.
00006  ABADDEED  F00F0006 22661050 0002247B ADC29CD9 4941DA55  N.A.
00007  ABADDEED  F00F0007 22661050 0003A47E 0994A026 2BADDAA0  N.A.
00008  ABADDEED  F00F0008 22661050 0003A47E 45A66BF7 E94525D5  N.A.
00009  ABADDEED  F00F0009 22661050 0004246F F40B9359 A7D41C19  N.A.

***** SUMMARY of DATA FORMAT ANALYSIS*****
File Name   : JPddcall.m5b
Data Format  : Mark-5B
File Size   : 534277415 bytes
Total # of Frames (estimated from file size): 53342.393670
1st Header Information
  Time (MJD HH:MM:SS.SSSS) : 226 16:57:30.0000
  Frame # (in a second)    : 0
  User Specified (16 bits) : F00F
  TVG data flag           : 0
Data Rate (Mbps) : 2048.0
Frames/sec       : 25600
Data Length (sec): 2.083687
*****
$

```

## 6 データフォーマット変換ソフトの使い方

### 6.1 k5tom5b

ユーティリティ名

k5tom5b

機能

K5 フォーマットから Mark5B フォーマットへの変換を行う

#### 6.1.1 実行方法

k5tom5b *k5name1* [*k5name2 k5name3 k5name4*] [*m5name*] [*options*]

ここで *k5name1*    – K5 データファイル名 (CH#01-04 または CH#01)  
*k5name2*    – K5 データファイル名 (CH#05-08 または CH#02)  
*k5name3*    – K5 データファイル名 (CH#09-12 または CH#03)  
*k5name4*    – K5 データファイル名 (CH#13-16 または CH#04)  
                   2 番目以降の K5 ファイルが 1 番目の K5 ファイルと同じディレクトリ  
                   にあるときはディレクトリを省略できる  
*m5name*    – 作成する Mark5B ファイル名  
**オプション (順不同)**  
 -o *m5name*    – 作成する Mark5B ファイル名  
 -d *m5dir*    – 作成する Mark5B ファイルのディレクトリ  
 -s *offset*    – データ変換開始秒 (データの先頭からのオフセットを秒単位で与える)  
                   デフォルトは 0  
 -p *period*    – データ変換を行うスパン (秒)。デフォルトはすべて  
 -e2bit *n*    – 2 ビットデータのエンコード方式  
                   *n*=1: オフセットバイナリ 0 1 2 3  
                   *n*=2: Mark-V 標準 0 2 1 3 (デフォルト)  
                   *n*=3: 符号付きバイナリ 1 0 2 3  
 -nocheck    – 2 番目以降のフレームデータのヘッダーエラーを無視する  
 -ch1 *nn*    – Mark5B データの ch#1 に相当する K5 データのチャンネル番号を設定する  
                   *nn* は K5 データのチャンネル番号 (1-16)  
 -ch2 *nn*    – Mark5B データの ch#2 に相当する K5 データのチャンネル番号を設定する  
                   .....  
 -ch16 *nn*    – Mark5B データの ch#16 に相当する K5 データのチャンネル番号を設定する  
 -chall *n1:n2: ... :n16*  
                   – Mark5B データの全チャンネルに相当する K5 データのチャンネル番号を設定する  
                   *n1* – Mark5B データの ch#1 に相当する K5 データのチャンネル番号 (1-16)

$n2$  – Mark5B データの ch#2 に相当する K5 データのチャンネル番号 (1-16)

.....

$n16$  – Mark5B データの ch#16 に相当する K5 データのチャンネル番号 (1-16)

#### [環境変数]

##### チェック法

k5tom5b env

M5DIR    – Mark5 データファイルのデフォルトディレクトリ  
M5VEX    – VEX スケジュールファイルのデフォルトディレクトリ

#### 6.1.2 実行例

```
$ k5tom5b ./tds.k5a tds.k5b tds.k5c tds.k5d data.m5bA data.m5b
```

```
*****
*   K5/VSSP to Mark-5B Data Format Converter   *
*   k5tom5b (Ver 1.70 2016-08-15) by T.KONDO/NICT *
*****
```

```
Second Length : 30.000000
Second Length : 30.000000
Second Length : 30.000000
Second Length : 30.000000
```

##### K5 file(s)

```
FILE1 CH# 1- 4 : ./tds.k5a
      (A/D(bits) 1 CHs 4 SFreq(kHz) 16000 Time 23:59:54 sec 86394) VSSP32
FILE2 CH# 5- 8 : ./tds.k5b
      (A/D(bits) 1 CHs 4 SFreq(kHz) 16000 Time 23:59:54 sec 86394) VSSP32
FILE3 CH# 9-12 : ./tds.k5c
      (A/D(bits) 1 CHs 4 SFreq(kHz) 16000 Time 23:59:54 sec 86394) VSSP32
FILE4 CH# 13-16 : ./tds.k5d
      (A/D(bits) 1 CHs 4 SFreq(kHz) 16000 Time 23:59:54 sec 86394) VSSP32
```

##### Mark5B file (created) : data.m5b

```
Observation Date : 3digit MJD = 589
                  :      Year = 2016
                  :      Month = 7
                  :      Day  = 20
                  :      Total Day = 202
Conversion length (sec) : 30
```

##### TIME LABEL MONITOR

```
=====
K5-FILE1 K5-FILE2 K5-FILE3 K5-FILE4 VLBA-TIME
HH:MM:SS HH:MM:SS HH:MM:SS HH:MM:SS JJJSSSSSSsss
-----
23:59:54 23:59:54 23:59:54 23:59:54 589863940000 OK
23:59:55 23:59:55 23:59:55 23:59:55 589863950000 OK
```

```

23:59:56 23:59:56 23:59:56 23:59:56 589863960000 OK
23:59:57 23:59:57 23:59:57 23:59:57 589863970000 OK
23:59:58 23:59:58 23:59:58 23:59:58 589863980000 OK
23:59:59 23:59:59 23:59:59 23:59:59 589863990000 OK
00:00:00 00:00:00 00:00:00 00:00:00 590000000000 OK
00:00:01 00:00:01 00:00:01 00:00:01 590000010000 OK
00:00:02 00:00:02 00:00:02 00:00:02 590000020000 OK
00:00:03 00:00:03 00:00:03 00:00:03 590000030000 OK

.....

00:00:19 00:00:19 00:00:19 00:00:19 590000190000 OK
00:00:20 00:00:20 00:00:20 00:00:20 590000200000 OK
00:00:21 00:00:21 00:00:21 00:00:21 590000210000 OK
00:00:22 00:00:22 00:00:22 00:00:22 590000220000 OK
00:00:23 00:00:23 00:00:23 00:00:23 590000230000 OK
=====

Mark5 file (data.m5b) has been created.

Time elapsed (sec) for One obs process is 52.000000
$

```

## 6.2 k5tom5

ユーティリティ名

k5tom5

機能

K5 フォーマットから Mark5 フォーマットへの変換を行う

### 6.2.1 How to execute

```

k5tom5 jjj|yyyymmdd|yyyddd k5name1 [k5name2 k5name3 k5name4 [m5name]] [options]
or
k5tom5 -i -infofile|make -v vexfile [options]
.... 情報ファイル作成モード

```

where *jjj|yyyymmdd|yyyddd*

– 観測日付の設定。以下のいずれかの方法で設定する

*jjj* : MJD(準ユリウス日) の下3桁

または

*yyyymmdd* : 4桁年, 2桁月, 2桁日

または

*yyyddd* : 4桁年, 3桁通日

*k5name1* – K5 データファイル名 (CH#01-04 または CH#01)

*k5name2* – K5 データファイル名 (CH#05-08 または CH#02)

*k5name3* – K5 データファイル名 (CH#09-12 または CH#03)

*k5name4* – K5 データファイル名 (CH#13-16 または CH#04)

2 番目以降の K5 ファイルが 1 番目の K5 ファイルと同じディレクトリにあるときはディレクトリを省略できる

*m5name*      -    作成する Mark5 ファイル名

オプション (順不同)

- i infofile*      -    以下の情報を含む情報ファイルの指定  
                                 トラック#とチャンネルの対応表  
                                 ビット位置とチャンネルの対応表  
                                 グループ#とチャンネルの対応表  
                                 (デフォルトファイル名は “k5tom5info.txt”)  
                                 “*-i -info\_file*” と名前の前に “-” をつけると *info\_file* という名前の情報  
                                 ファイルが作成される。  
                                 “*-i make*” とするとデフォルトの情報ファイルが作成される。
- o m5name*      -    作成する Mark5 ファイル名
- d m5dir*        -    作成する Mark5 ファイルのディレクトリ
- s soffset*      -    データ変換開始秒 (データの先頭からのオフセットを秒単位で与える)  
                                 デフォルトは 0
- p period*       -    データ変換を行うスパン (秒)。デフォルトはすべて
- vlba*           -    フレームフォーマットを VLBA フォーマットにする (デフォルトは  
                                 Mark4 フォーマット)
- long*           -    64 ビット (トラック) モードにする (デフォルトは 32 ビットモード)
- v vex\_file*     -    情報ファイル作成モード時に読み込む VEX ファイルを指定する
- sid stat\_id*    -    情報ファイル作成モード時に情報を取得する局 ID を指定する
- scan scan#*    -    情報ファイル作成モード時参照するスキャン#を指定する (デフォルト  
                                 は 1)
- short*          -    Mark5 ファイルのスキャン長を K5 ファイルのスキャン長の最小値に  
                                 セットする (デフォルトは最長値)。データの無いところには 0 データ  
                                 が埋め込まれる。
- monit*          -    実行状況をモニターする
- nocheck*       -    2 番目のフレーム以降のヘッダーエラーを無視する

#### [環境変数]

##### チェック法

k5tom5 env

- M5DIR      -    Mark5 データファイルのデフォルトディレクトリ
- M5VEX      -    VEX スケジュールファイルのデフォルトディレクトリ

#### 6.2.2 実行例

例 1: デフォルト変換情報ファイルを使っての変換

データ日付を年と通日でセットしデフォルト情報ファイル “k5tom5info.txt” を使用する

```
k5tom5 20031015 02880001.k5a 02880001.k5b 02880001.k5c 02880001.k5d m5test.dat
```

```
*****
*      K5/VSSP to Mark-V Data Format Converter      *
*      k5tom5 (Ver 1.21 2005-09-17) by T.KONDO/NICT  *
*****
```

```
----- RUN CONDITION -----
```

```
yyyyymmdd : 20031015
yyyyddd   : 2003288
jjj       : 927
MJD       : 52927
k5name 1  : 02880001.k5a
k5name 2  : 02880001.k5b
k5name 3  : 02880001.k5c
k5name 4  : 02880001.k5d
mk5name   : m5test.dat2
outdir    :
infofile  : k5tom5info.txt
channel   : 0
group     : 0
soffset   : 0
period    : 0
odd       : 0
parity    : 0
nrzm      : -1
vlba      : 0
nbits     : 32
-----
```

K5 file(s)

```
FILE1 CH# 1- 4 : 02880001.k5a
  (A/D(bits) 1 CHs 4 SFreq(kHz) 4000 Time 17:30:02 sec 63002)
FILE2 CH# 5- 8 : 02880001.k5b
  (A/D(bits) 1 CHs 4 SFreq(kHz) 4000 Time 17:30:02 sec 63002)
FILE3 CH# 9-12 : 02880001.k5c
  (A/D(bits) 1 CHs 4 SFreq(kHz) 4000 Time 17:30:02 sec 63002)
FILE4 CH# 13-16 : 02880001.k5d
  (A/D(bits) 1 CHs 4 SFreq(kHz) 4000 Time 17:30:02 sec 63002)
```

Mark5 file (created) : m5test.dat2

```
Observation Date : 3digit MJD = 927
                  :      Year = 2003
                  :      Month = 10
                  :      Day  = 15
                  : Total Day = 288
```

#### TIME LABEL MONITOR

```
=====
K5-FILE1 K5-FILE2 K5-FILE3 K5-FILE4 MarkIV-TIME
HH:MM:SS HH:MM:SS HH:MM:SS HH:MM:SS YDDHHMMSSsss
-----
17:30:02 17:30:02 17:30:02 17:30:02 3288173002000 OK
17:30:03 17:30:03 17:30:03 17:30:03 3288173003000 OK
17:30:04 17:30:04 17:30:04 17:30:04 3288173004000 OK
17:30:05 17:30:05 17:30:05 17:30:05 3288173005000 OK
17:30:06 17:30:06 17:30:06 17:30:06 3288173006000 OK
17:30:07 17:30:07 17:30:07 17:30:07 3288173007000 OK
17:30:08 17:30:08 17:30:08 17:30:08 3288173008000 OK
17:30:09 17:30:09 17:30:09 17:30:09 3288173009000 OK
17:30:10 17:30:10 17:30:10 17:30:10 3288173010000 OK
```

```
17:30:11 17:30:11 17:30:11 17:30:11 3288173011000 OK
=====
```

Mark5 file (m5test.dat2) has been created.

Time elapsed (sec) for One obs process is 44.000000

## 例 2: 変換情報ファイルを作成する

VEX ファイル “/home/vlbi/mark5/gg057c.vex” を使用して情報ファイルを作成する。参照する局 ID は “Ks”。

```
k5tom5 -i -/home/vlbi/mark5/k5tom5info.txt -v /home/vlbi/mark5/gg057c.vex -sid Ks
K5tom5 running under Information File create mode
info file (/home/vlbi/mark5/k5tom5info.txt) will be created (updated).
VEX file name --- /home/vlbi/mark5/gg057c.vex
SITES (Station ID) defined are
```

ID	SITE NAME
Pt	VLBA_PT
Kp	VLBA_KP
La	VLBA_LA
Br	VLBA_BR
Fd	VLBA_FD
Nl	VLBA_NL
Ov	VLBA_OV
Mk	VLBA_MK
Gb	GBT_VLBA
Ks	KASHIM34
At	ATCA
Sh	SHANGHAI
Mp	MOPRA
Cd	CEDUNA
Ho	HOBART
Ur	URUMQI
Pa	PARKES5

```
-----
Selected Station ID = Ks
search_site: No CLOCK info for KASHIM34 included in the VEX FILE.
search_site: So all 0 for clock information was set.
```

```
Detailed site information
  site definition : KASHIM34
  site name      : KASHIM34
  site ID       : Ks
  site position  : -3997649.222000 3276690.753000 3724278.823000
  site clock
    validity epoch : 0 0 0 0 0
    clock epoch    : 0 0 0 0 0
    clock offset   : 0.000000e+000
    clock rate     : 0.000000e+000
  Scan # for mode get is 1
mode is huygS
Mode was taken from Scan #1 as huygS
```

```
TRACK and FREQUENCY information for KASHIM34
BARREL ROLL : off
FREQDEF = 2034.99MHz8x16MHz TRACKDEF = MKIV.8Ch2bit1to4
adbit= 2 sample_rate= 32000000.000000
```

bb	HS	Tr	AD	fo	chan	bbcid	BBC#	RF(Hz)	S	VBW(Hz)	PASS
1	1	2	sign	1	&CH01 &BBC01	1	1	2034990000.0	L	16000000.0	
2	1	4	sign	2	&CH01 &BBC01	1	1	2034990000.0	L	16000000.0	
3	1	6	sign	3	&CH01 &BBC01	1	1	2034990000.0	L	16000000.0	



```

 4  1  8  sign  4  &CH01 &BBC01  1  2034990000.0 L  16000000.0
 5  1 10  mag  1  &CH01 &BBC01  1  2034990000.0 L  16000000.0
 6  1 12  mag  2  &CH01 &BBC01  1  2034990000.0 L  16000000.0
 7  1 14  mag  3  &CH01 &BBC01  1  2034990000.0 L  16000000.0
 8  1 16  mag  4  &CH01 &BBC01  1  2034990000.0 L  16000000.0
 9  1 18  sign  1  &CH02 &BBC02  2  2034990000.0 L  16000000.0
10  1 20  sign  2  &CH02 &BBC02  2  2034990000.0 L  16000000.0
11  1 22  sign  3  &CH02 &BBC02  2  2034990000.0 L  16000000.0
12  1 24  sign  4  &CH02 &BBC02  2  2034990000.0 L  16000000.0
13  1 26  mag  1  &CH02 &BBC02  2  2034990000.0 L  16000000.0
14  1 28  mag  2  &CH02 &BBC02  2  2034990000.0 L  16000000.0
15  1 30  mag  3  &CH02 &BBC02  2  2034990000.0 L  16000000.0
16  1 32  mag  4  &CH02 &BBC02  2  2034990000.0 L  16000000.0
17  1  3  sign  1  &CH03 &BBC01  1  2034990000.0 U  16000000.0
18  1  5  sign  2  &CH03 &BBC01  1  2034990000.0 U  16000000.0
19  1  7  sign  3  &CH03 &BBC01  1  2034990000.0 U  16000000.0
20  1  9  sign  4  &CH03 &BBC01  1  2034990000.0 U  16000000.0
21  1 11  mag  1  &CH03 &BBC01  1  2034990000.0 U  16000000.0
22  1 13  mag  2  &CH03 &BBC01  1  2034990000.0 U  16000000.0
23  1 15  mag  3  &CH03 &BBC01  1  2034990000.0 U  16000000.0
24  1 17  mag  4  &CH03 &BBC01  1  2034990000.0 U  16000000.0
25  1 19  sign  1  &CH04 &BBC02  2  2034990000.0 U  16000000.0
26  1 21  sign  2  &CH04 &BBC02  2  2034990000.0 U  16000000.0
27  1 23  sign  3  &CH04 &BBC02  2  2034990000.0 U  16000000.0
28  1 25  sign  4  &CH04 &BBC02  2  2034990000.0 U  16000000.0
29  1 27  mag  1  &CH04 &BBC02  2  2034990000.0 U  16000000.0
30  1 29  mag  2  &CH04 &BBC02  2  2034990000.0 U  16000000.0
31  1 31  mag  3  &CH04 &BBC02  2  2034990000.0 U  16000000.0
32  1 33  mag  4  &CH04 &BBC02  2  2034990000.0 U  16000000.0
33  2  2  sign  1  &CH05 &BBC03  3  2139990000.0 L  16000000.0
34  2  4  sign  2  &CH05 &BBC03  3  2139990000.0 L  16000000.0
35  2  6  sign  3  &CH05 &BBC03  3  2139990000.0 L  16000000.0
36  2  8  sign  4  &CH05 &BBC03  3  2139990000.0 L  16000000.0
37  2 10  mag  1  &CH05 &BBC03  3  2139990000.0 L  16000000.0
38  2 12  mag  2  &CH05 &BBC03  3  2139990000.0 L  16000000.0
39  2 14  mag  3  &CH05 &BBC03  3  2139990000.0 L  16000000.0
40  2 16  mag  4  &CH05 &BBC03  3  2139990000.0 L  16000000.0
41  2 18  sign  1  &CH06 &BBC03  3  2139990000.0 U  16000000.0
42  2 20  sign  2  &CH06 &BBC03  3  2139990000.0 U  16000000.0
43  2 22  sign  3  &CH06 &BBC03  3  2139990000.0 U  16000000.0
44  2 24  sign  4  &CH06 &BBC03  3  2139990000.0 U  16000000.0
45  2 26  mag  1  &CH06 &BBC03  3  2139990000.0 U  16000000.0
46  2 28  mag  2  &CH06 &BBC03  3  2139990000.0 U  16000000.0
47  2 30  mag  3  &CH06 &BBC03  3  2139990000.0 U  16000000.0
48  2 32  mag  4  &CH06 &BBC03  3  2139990000.0 U  16000000.0
49  2  3  sign  1  &CH07 &BBC04  4  2286990000.0 L  16000000.0
50  2  5  sign  2  &CH07 &BBC04  4  2286990000.0 L  16000000.0
51  2  7  sign  3  &CH07 &BBC04  4  2286990000.0 L  16000000.0
52  2  9  sign  4  &CH07 &BBC04  4  2286990000.0 L  16000000.0
53  2 11  mag  1  &CH07 &BBC04  4  2286990000.0 L  16000000.0
54  2 13  mag  2  &CH07 &BBC04  4  2286990000.0 L  16000000.0
55  2 15  mag  3  &CH07 &BBC04  4  2286990000.0 L  16000000.0
56  2 17  mag  4  &CH07 &BBC04  4  2286990000.0 L  16000000.0
57  2 19  sign  1  &CH08 &BBC04  4  2286990000.0 U  16000000.0
58  2 21  sign  2  &CH08 &BBC04  4  2286990000.0 U  16000000.0
59  2 23  sign  3  &CH08 &BBC04  4  2286990000.0 U  16000000.0
60  2 25  sign  4  &CH08 &BBC04  4  2286990000.0 U  16000000.0
61  2 27  mag  1  &CH08 &BBC04  4  2286990000.0 U  16000000.0
62  2 29  mag  2  &CH08 &BBC04  4  2286990000.0 U  16000000.0
63  2 31  mag  3  &CH08 &BBC04  4  2286990000.0 U  16000000.0
64  2 33  mag  4  &CH08 &BBC04  4  2286990000.0 U  16000000.0
default Mark-V data format is as follows
data encode -- NRZL without parity
data format -- Mark-IV
#_of_track  = 64

```

```
Information file (/home/vlbi/mark5/k5tom5info.txt) created!!
```

### 6.3 k5tovdif

ユーティリティ名

k5tovdif

機能

K5 フォーマットから VDIF フォーマットへの変換を行う

#### 6.3.1 実行方法

```
k5tovdif k5name1 [k5name2 k5name3 k5name4 [vdifname]] [options]
```

- where
- k5name1*        – K5 データファイル名 (CH#01-04 または CH#01)
  - k5name2*        – K5 データファイル名 (CH#05-08 または CH#02)
  - k5name3*        – K5 データファイル名 (CH#09-12 または CH#03)
  - k5name4*        – K5 データファイル名 (CH#13-16 または CH#04)
- 2 番目以降の K5 ファイルが一番目の K5 ファイルと同じディレクトリにあるときはディレクトリを省略できる
- vdifname*        – 作成する VDIF ファイル名
- オプション (順不同)
- o *vdifname*     – 作成する VDIF ファイル名 (デフォルトは最初の K 5 ファイル名の拡張子を ‘.vdif’ で置き換えた名前)
  - dir *vdifdir*    – 作成する VDIF ファイルのディレクトリ (デフォルトは K5 ディレクトリ)
  - so *soffset*     – データ変換開始秒 (データの先頭からのオフセットを秒単位で与える) デフォルトは 0
  - p *period*       – データ変換を行うスパン (秒)。デフォルトはすべて
  - das *das\_name*   – 局名をセットする (K5/VSSP32 フォーマットの場合はヘッダーの AUX フィールド中の局名またはホスト PC 名を使う)
  - sid *sid*        – 局 ID(A2) をセットする (K5/VSSP32 フォーマットの場合はヘッダーの AUX フィールド中から取得する)
  - date *jjj|yyyymmdd|yyyyddd*
    - K5/VSSP フォーマットの場合は日付情報が無いので日付を以下のよう
    - にセットする
    - jjj* : MJD(準ユリウス日) の下 3 桁
    - または
    - yyyymmdd* : 4 桁年, 2 桁月, 2 桁日
    - または
    - yyyyddd* : 4 桁年, 3 桁通日
  - e2bit *n*        – 2 ビットデータのエンコード方式
- n*=1 : オフセットバイナリ

```

n=2 : Mark-V 標準 0 2 1 3 (デフォルト)
n=3 : 符号付きバイナリ 1 0 2 3
-nocheck      - 2 番目以降のフレームデータのヘッダーエラーを無視する
-fr[amlength] frame_bytes
               - ヘッダー部を除くフレーム長をセットする (バイト) (デフォルトは 1280)
-num[frame_sec] nframe
               - 1 秒中のフレーム数をセットする (デフォルトは  $2048 \times 10^6 / (1280) = 200000$ )
-ch1 nn      - VDIF データの ch#1 に相当する K5 データのチャンネル番号を設定する
               nn は K5 データのチャンネル番号 (1-16)
-ch2 nn      - VDIF データの ch#2 に相当する K5 データのチャンネル番号を設定する
               .....
-ch16 nn     - VDIF データの ch#16 に相当する K5 データのチャンネル番号を設定する
               .....
-chall n1:n2: ... :n16
               - VDIF データの全チャンネルに相当する K5 データのチャンネル番号を設定する
               n1 - VDIF データの ch#1 に相当する K5 データのチャンネル番号 (1-16)
               n2 - VDIF データの ch#2 に相当する K5 データのチャンネル番号 (1-16)
               .....
               n16 - VDIF データの ch#16 に相当する K5 データのチャンネル番号 (1-16)

```

## [環境変数]

## チェック法

k5tovdif env

VDIFDIR - VDIF データファイルのデフォルトディレクトリ

## 6.4 ads2k5

## ユーティリティ名

ads2k5

## 機能

ADS3000 (DBBC モード) フォーマットから K5 フォーマットへの変換を行う

## 6.4.1 実行方法

```
ads2k5 adsname[options]
```

where	<i>adsname</i>	–	ADS フォーマットデータファイル名
	オプション (順不同)		
	<i>-c channel</i>	–	1 チャンネル変換時の抽出チャンネル番号 (1-16) ('-1ch' オプションと同じ。デフォルトは 4x4ch モード)
	<i>-u unit</i>	–	4 チャンネルモード時の抽出ユニット (グループ) 番号 (1-4) ('-c' オプションはこのオプションより優先される。デフォルトは 4x4ch モード)
	<i>-o k5name</i>	–	作成される K5 ファイル名 (デフォルトは以下の命名則参照)
	<i>-d k5dir</i>	–	K5 ファイル出力ディレクトリの設定
	<i>-fsampl fMHz</i>	–	サンプリング周波数 (MHz) (デフォルトは 32)
	<i>-adbit adbit</i>	–	AD ビット数 (デフォルトは 1)
	<i>-1ch ch</i>	–	作成される K5 データを 1 チャンネルモードにして抽出する ADS データのチャンネル番号を指定する (チャンネル番号は 1 から始まる)
	<i>-4ch ch1 ch2 ch3 ch4</i>	–	作成される K5 データを 4 チャンネルモードにして抽出する ADS データのチャンネル番号を指定する (デフォルトは 4ch モードの K5 ファイルを 4 つ作成)
	<i>-2bit mode</i>	–	2 ビットデータのデコード方式の選択 1 : オフセットバイナリ 0 1 2 3 (デフォルト) 2 : Mark-V 標準 0 2 1 3 3 : 符号付きバイナリ 1 0 2 3
	<i>-t YYYYDDDDHHMMSS YYYY/DDD-HH:MM:SS</i> または <i>HHMMSS HH:MM:SS</i> または <i>YYYYDDD YYYY/DDD</i>	–	データ開始日時をセットする。省略時は ADS データファイル名から情報を取得する
	<i>-s offset</i>	–	データ変換開始秒 (データの先頭からのオフセットを秒単位で与える) デフォルトは 0
	<i>-p period</i>	–	データ変換を行うスパン (秒)。デフォルトはすべて
	<i>-vssp</i>	–	K5 フォーマットを K5/VSSP に設定する (デフォルトは K5/VSSP32 フォーマット)
	<i>-monit</i>	–	実行状況をモニターする
	<i>-adsbit adsbit</i>	–	ADS データの AD ビットを設定する (2 4 8) (デフォルトは 2) もし <i>adbit</i> $\geq 4$ なら <i>adsbit</i> = <i>adbit</i>
	<i>-adsnumch adsnumch</i>	–	ADS データのチャンネル数を設定する (1 8) (デフォルトは 8)

[作成される K5 データファイルの命名則]

(ADS : ADS3000 ファイル名または '-o' オプションで与えられる名前とする)

## 4x4ch モード

ADS.k5a ---- グループ#1 (ch01-04)

ADS.k5b ---- グループ#2 (ch05-08)

ADS.k5c ---- グループ#3 (ch09-12)

ADS.k5d ---- グループ#4 (ch13-16)

## 4ch モード

ADS.k5[a|b|c|d]

## 1ch モード

ADS.k5-NN ---- NN はチャンネル番号 01-16

## [環境変数]

## チェック法

ads2k5 env

K5_CH_TABLE	–	オペレータが設定するチャンネル配置表
ADSDIR	–	ADS データファイルのデフォルトディレクトリ
K5DIR	–	K5 データファイルのデフォルトディレクトリ

## 注意:

もし ADS データファイル名が以下の条件を満たすときは開始日時、サンプリング情報をファイル名から得ることが可能である。

1. ファイル名が以下の構造をしている。

xxx...xxxx\_YYYYDDDDHHMMSS.raw

ここで xxx...xxxx — 任意の文字列

YYYYDDDDHHMMSS – データ開始時刻 (年, 通日, 時, 分, 秒)

2. “xxx...xxxx” の部分は “16MHz” や “8Msps” のようにサンプリング情報を含む (大文字、小文字の区別なし)
3. このサンプリング情報部の前の部分は非数字のアスキー文字列またはファイル名の開始である。例、  
...dbbc\_16MHz\_..... または ...xyz.32MSPS.xxxxxxx

## 6.5 m5btok5

## ユーティリティ名

m5btok5

## 機能

Mark5B フォーマットから K5 フォーマットへの変換を行う

## 6.5.1 実行方法

m5btok5 m5bname [options]

```

where  m5bname      -   Mark5B フォーマットデータファイル
        オプション (順不同)
        -o k5name    -   作成される K5 ファイル名 (デフォルトは以下の命名則参照)
        -d k5dir     -   K5 ファイル出力ディレクトリの設定
        -bs nbitsream -   Mark5B データのビットストリーム値を設定する (1,2,4,8,16,32)
        -fsampl fmhz -   サンプリング周波数 (MHz) (デフォルトは 4)
        -adbit adbit  -   AD ビット数 (デフォルトは 1)
        -1ch ch      -   作成される K5 データを 1 チャンネルモードにして抽出する Mark5B
                        データのチャンネル番号を指定する (チャンネル番号は 1 から始まる)
        -4ch ch1 ch2 ch3 ch4
                        -   作成される K5 データを 4 チャンネルモードにして抽出する Mark5B
                        データのチャンネル番号を指定する
        -2bit mode    -   2 ビットデータのデコード方式の選択
                        1 : オフセットバイナリ 0 1 2 3
                        2 : Mark-V 標準 0 2 1 3 (デフォルト)
                        3 : 符号付きバイナリ 1 0 2 3

```

[作成される K5 データファイルの命名則]

(MK5 : Mrak5B データファイル名)

4ch モード

```

MK5.k5a ---- グループ#1 (ch01-04)
MK5.k5b ---- グループ#2 (ch05-08)
MK5.k5c ---- グループ#3 (ch09-12)
MK5.k5d ---- グループ#4 (ch13-16)

```

1ch モード

```

MK5.k5-NN ---- NN はチャンネル番号 01-16

```

[環境変数]

チェック法

```

m5btok5 env

```

```

M5DIR  -   Mark5 データファイルのデフォルトディレクトリ
K5DIR  -   K5 データファイルのデフォルトディレクトリ

```

## 6.6 m5tok5

ユーティリティ名

m5tok5

機能

Mark5 フォーマットから K5 フォーマットへの変換を行う

### 6.6.1 実行方法

モード 1

m5tok5 *m5name* [*options*]

where *m5name*            – Mark5 データファイル名

オプション (順不同)

- c *channel*           – 1 チャンネルモード時の抽出チャンネル番号 (1-16) (デフォルトは 4x4ch モード)
- g *group*            – 4 チャンネルモード時の抽出ユニット (グループ) 番号 (1-4) ('-c' オプションはこのオプションより優先される。デフォルトは 4x4ch モード)
- i *infofile*          – 以下の情報を含む情報ファイルの指定  
                         トラック#とチャンネルの対応表  
                         ビット位置とチャンネルの対応表  
                         グループ#とチャンネルの対応表  
                         (デフォルトファイル名は “m5tok5info.txt”)  
                         “-i -info\_file” と名前の前に “-” をつけると *info\_file* という名前の情報ファイルが作成される。  
                         “-i make” とするとデフォルトの情報ファイルが作成される。
- o *k5name*           – 作成される K5 ファイル名 (デフォルトは以下の命名則参照)
- d *k5dir*            – K5 ファイル出力ディレクトリの設定 (デフォルトは Mark5 データディレクトリ)
- s *soffset*          – データ変換開始秒 (データの先頭からのオフセットを秒単位で与える) デフォルトは 0
- p *period*           – データ変換を行うスパン (秒)。デフォルトはすべて
- r                    – fanout=2 またそれ以上の時、トラックの配列を逆にする
- v *vex\_file*          – 情報ファイル作成モード時に読み込む VEX ファイルを指定する
- sid *stat\_id*        – 情報ファイル作成モード時に情報を取得する局 ID を指定する
- scan *scan#*         – 情報ファイル作成モード時参照するスキャン#を指定する (デフォルトは 1)
- subp *subpass*       – 情報ファイルのサブパス情報をセットする (デフォルトは “A”)
- odd                  – Mark5 データの読み込み開始位置を 32bit (4byte) シフトする (64 トラックモード時のみ有効)
- monit                – 実行状況をモニターする

## モード 2

```
m5tok5 m5name k5file bit1 [bit2 bit3 .. bitN] [options2]
```

where *m5name*        - Mark5 データファイル名  
*k5file*            - 作成される K5 ファイル名  
*bit1*              - 最初に抽出するビット位置 (0-トラック数)  
*bit2*              - 2 番目に抽出するビット位置 (0-トラック数)  
.....  
*bitN*             - N 番目に抽出するビット位置 (0-トラック数)  
                    注: *N* は (K5 チャンネル数)\*(AD ビット数)\*(Mark5 ファンアウト)

## オプション 2 (順不同)

-track *ntrack*    - Mark5 データのトラック数 (8,16,32,64) (デフォルトは 32)  
-fsampl *fmhz*     - サンプリング周波数 (MHz) (デフォルトは 4)  
-adbit *adbit*     - AD ビット数 (デフォルトは 1)  
-nch *numch*       - K5 フォーマットでのチャンネル数 (1 または 4) (デフォルトは 4)  
-parity            - Mark5 データはパリティあり  
-noparity          - Mark5 データはパリティなし (デフォルト)  
-vlba             - VLBA モードに設定する (デフォルトは MarkIV モード)  
-s *offset*        - データ変換開始秒 (データの先頭からのオフセットを秒単位で与える)  
                    デフォルトは 0  
-p *period*        - データ変換を行うスパン (秒)。デフォルトはすべて  
-odd               - Mark5 データの読み込み開始位置を 32bit (4byte) シフトする (64 トラックモード時のみ有効)

## [作成される K5 データファイルの命名則]

(MK5 : Mrak5B データファイル名)

## 4ch モード

MK5.k5a ---- グループ#1 (ch01-04)  
MK5.k5b ---- グループ#2 (ch05-08)  
MK5.k5c ---- グループ#3 (ch09-12)  
MK5.k5d ---- グループ#4 (ch13-16)

## 1ch モード

MK5.k5-NN ---- NN はチャンネル番号 01-16

## [環境変数]

## チェック法

```
m5tok5 env
```



- M5DIR    –   Mark5 データファイルのデフォルトディレクトリ
- M5VEX    –   VEX ファイルのデフォルトディレクトリ

## 6.7 vdif2k5

ユーティリティ名

vdif2k5

機能

VDIF フォーマットから K5 フォーマットへの変換を行う

### 6.7.1 実行方法

vdif2k5 *vdiffile* [*options*]

- where *vdiffile*        –   VDIF フォーマットデータファイル
- オプション (順不同)
- o *k5name*        –   作成される K5 ファイル名 (デフォルトは以下の命名則参照)
  - d *k5dir*        –   K5 ファイル出力ディレクトリの設定。-o オプションで指定した K5 ファイル名がディレクトリを含む場合はこのオプションは無効
  - fsampl *fmhz*    –   サンプリング周波数 (MHz) (デフォルトは 4)
  - adbit *adbit*    –   AD ビット数 (デフォルトは 1)
  - 1ch *ch*         –   作成される K5 データを 1 チャンネルモードにして抽出する VDIF データのチャンネル番号を指定する (チャンネル番号は 1 から始まる)
  - 4ch *ch1 ch2 ch3 ch4*        –   作成される K5 データを 4 チャンネルモードにして抽出する VDIF データのチャンネル番号を指定する
  - 2bit *mode*      –   2 ビットデータのデコード方式の選択
    - 1 : オフセットバイナリ 0 1 2 3 (デフォルト)
    - 2 : Mark-V 標準 0 2 1 3
    - 3 : 符号付きバイナリ 1 0 2 3
  - s *offset*       –   データ変換開始秒 (データの先頭からのオフセットを秒単位で与える)
    - デフォルトは 0
    - 負の値の場合は絶対値が最初に読み飛ばすバイト数となる

[作成される K5 データファイルの命名則]

- 1) -o オプションを使わない場合 (K5 ファイル名を指定しない場合)

## 4ch モード

VDIF.k5a ---- グループ#1 (ch01-04)

VDIF.k5b ---- グループ#2 (ch05-08)

VDIF.k5c ---- グループ#3 (ch09-12)

VDIF.k5d ---- グループ#4 (ch13-16)

## 1ch モード (複数チャンネル VDIF データから変換)

MK5.k5-NN ---- NN はチャンネル番号 01-16

## 1ch モード (1ch モードの VDIF データから変換)

VDIF.k5

(VDIF : VDIF データファイル名)

## 2) -o オプションを使った場合 (K5 ファイル名を指定した場合)

## 出力ファイルが 1 つのみの場合

k5name --- '-o' オプションで与えられた名前

## 出力ファイルが複数 ( 2 つ ) の場合

k5name.1 k5name.2

## 出力ファイルが複数 ( 4 つ ) の場合

k5name.1 k5name.2 k5name.3 k5name.4

## 6.8 vdif2m5b

## ユーティリティ名

vdif2m5b

## 機能

VDIF フォーマットから Mark5B フォーマットへの変換を行う

## 6.8.1 実行方法

vdif2m5b *vdiffile* [*options*]where *vdiffile* - VDIF フォーマットデータファイル

## オプション (順不同)

-o *m5bname* - 作成される Mark5B ファイル名 (デフォルトは以下の命名則参照)-d *m5dir* - Mark5B ファイル出力ディレクトリの設定 (デフォルトは VDIF データディレクトリ)-fsampl *fmhz* - サンプリング周波数 (MHz) (デフォルトは 4)-2bit *mode* - 2 ビットデータのデコード方式の選択

1 : オフセットバイナリ 0 1 2 3

2 : Mark-V 標準 0 2 1 3 (デフォルト)

3 : 符号付きバイナリ 1 0 2 3

-s *soffset* - データ変換開始秒 (データの先頭からのオフセットを秒単位で与える) デフォルトは 0-p *period* - データ変換を行うスパン (秒)。デフォルトはすべて

- type1            – 出力ファイルの命名を Type 1 で行う (ファイル拡張子を “m5b” で置き換える)
- pid *thread\_id* – 抽出する VDIF データのスレッド ID を強制的にセットする
- ch1 *nn*         – Mark5B ch#1 に相当する VDIF データチャンネル番号をセットする  
*nn* は VDIF チャンネル番号 (1-16)
- ch2 *nn*         – Mark5B ch#2 に相当する VDIF データチャンネル番号をセットする
- .....
- ch16 *nn*        – Mark5B ch#16 に相当する VDIF データチャンネル番号をセットする
- chall *n1:n2: ... :n16*
  - Mark5B の全チャンネルに相当する VDIF データのチャンネル番号をセットする
  - n1* – Mark5B CH#1 に相当する VDIF チャンネル番号 (1-16)
  - n2* – Mark5B CH#2 に相当する VDIF チャンネル番号 (1-16)
  - .....
  - n16* – Mark5B CH#16 に相当する VDIF チャンネル番号 (1-16)

[作成される Mark5B データファイルの命名則]

(VDIF.vdif : VDIF データファイル名)

Type 0   VDIF.vdif.m5b (デフォルト)  
Type 1   VDIF.m5b

## 7 相関処理の実際

相関処理は

### 1. フリンジサーチ

基線ごとのクロックオフセット・レートを決める

### 2. 全観測（スキャン）の処理

1 で決定したクロックオフセット・レートを使って全観測（スキャン）の処理を行う

の 2 段階に分かれる。

### 7.1 フリンジサーチ

測地 VLBI 実験の定常相関処理は効率よく処理を行うため遅延ラグ数（サンプリング周期単位）を小さく（32 が一般的である）とすることが多い。このラグ範囲の中央付近に相関のピークを持ってくる必要があり、そのためには 2 局間の時刻同期誤差（クロックオフセット）の精密な値（少なくともサンプリング周期の数分の 1 程度の精度）が必要になる。

フリンジサーチとは VLBI 観測データ（サンプリングデータ）の相互相関処理を行ない相関を検出してクロックオフセットを決定する作業である。相関処理を行うには 2 局に到達する電波の時間差（遅延時間）の予測値が必要であるが、この遅延時間は地球回転によって時々刻々と変化する。この予測値は局位置と電波源の位置情報およびクロックオフセットから計算によって求められる<sup>1</sup>が、2 局間の時刻同期誤差（クロックオフセット）は事前に精度よく求めることが難しい。そこで、クロックオフセットとして“仮の値”を用いて予測値を計算し、その予測値に基づくフリンジサーチによりクロックオフセットを決定する。クロックオフセットの“仮の値”としては各局の時計と UTC の比較が可能な場合（通常 GPS が用いられる）はその値を、比較手段がない場合は 0 を用いる。

測地実験のように実験（セッション<sup>2</sup>）が 24 時間のような長時間に及ぶ場合にはクロックレートも求めてやる。フリンジサーチおよびクロックレート決定の具体的手順を以下に記す。

#### 1. 処理する観測（スキャン）を決める

- クロックレートを求めない場合：セッションの真中辺りの強い電波源を観測したスキャン
- クロックレートを求める場合：セッションの最初と最後の辺りの強い電波源を観測したスキャン

強い電波源のスキャン番号を調べるにはスケジュールファイルの中身をテキストエディタで見る必要があるが、Ver.2016-10-12 以降の apri\_calc であれば電波源や基線を指定したスケジュールモニターモードで起動すると以下のように簡単に調べることができる（以後、apri\_calc のバージョンは 2016-10-12 以降の使用を仮定）。ここではスケジュールファイル名を sample.skd とし sample.skd のあるディレクトリを作業ディレクトリとする。

#### (a) apri\_calc をモニターモードで実行する

```
apri_calc -sample.skd
```

または

```
apri_calc sample.skd -monit
```

スケジュールファイル中の電波源リストおよび参加局 ID のリストを見ることができるので、強い電波源を選ぶ（3C273B, 3C290, 4C39.25 など）。

注：スケジュールファイル名の前に“-”をつけるとモニターモードとなる。スケジュールファイル名を例えば /home/vlbi/sked/sample.skd のようにパス名を含めたいときはパス名の先頭に“-”をつけて

apri\_calc -/home/vlbi/sked/sample.skd のように起動する（ではあるが“-monit”オプションの使用を推奨）。

<sup>1</sup>地球姿勢パラメータも予測値に影響を及ぼすが実用的には 0 として大丈夫である

<sup>2</sup>数分の観測をスキャンと呼び、スキャンの集まりをセッションと呼ぶ

(b) 続いて apri\_calc を電波源を指定したモニターモードで実行する

```
apri_calc -sample.skd -source 3C273B
```

または

```
apri_calc sample.skd -monit -source 3C273B
```

指定した電波源のスキャンのみが表示されるので、スキャン番号をメモする。基線を絞る場合は参加局 ID のリストから”-baseid” オプションを使って基線を指定して

```
apri_calc sample.skd -monit -source 3C273B -basei RY
```

のように実行する。すると以下のような情報が得られる。

```
SkdMonit: ----- PICKUP SCAN TABLE -----
SkdMonit: SCAN#      SOURCE YYYY/DDD HH:MM:SS DURA STATION_IDS
SkdMonit:      8      3C273B 2003/197 02:40:20 100 T A C J R Y H
SkdMonit:     17      3C273B 2003/197 03:56:00 100 T A C J R Y H
SkdMonit:     27      3C273B 2003/197 04:54:40 100 T A C J R Y H K
SkdMonit:     35      3C273B 2003/197 05:54:10 100 T A C J R Y H
SkdMonit:     44      3C273B 2003/197 06:51:40 100 T A C J R Y H
SkdMonit:     53      3C273B 2003/197 07:53:10 100 A C J R Y H
SkdMonit:     71      3C273B 2003/197 10:11:40 100 A C J R Y H
SkdMonit:     81      3C273B 2003/197 11:11:00 100 T A C J R Y H
SkdMonit:     94      3C273B 2003/197 12:31:30 100 T A C J R Y H K
SkdMonit: -----
```

SCAN#で示されている番号が抽出されたスキャン番号である。

## 2. 予測値の計算

1 で決めたスキャンに対して予測値計算ソフト apri\_calc を使用して予測値を計算する。1 で抽出される全スキャン番号の予測値を計算したいなら、以下のように実行しても良い

```
apri_calc sample.skd -source 3C273B -basei RY
```

この場合は予測値ファイル出力ディレクトリ等はデフォルト値を使用しているが、他のオプション指定については apri\_calc の使用方法 (3.1 章) を参照されたい。

予測値ファイルはテキストファイルなので (フォーマットの詳細は付録 A 参照のこと)、必要な場合 (例えば生データファイル名やディレクトリの変更) はテキストエディタで簡単に修正できる。

## 3. 相関処理の実行

fx\_cor (K5/VSSP 以外のデータの場合は fx\_cor\_new) を用いてラグ数を大きく取って (1024 またはそれ以上) 相関処理を実行する。具体的に予測値ファイルを apesample.txt とし予測値ファイルのあるディレクトリで作業を行うとすると以下のように実行する

```
fx_cor ./apesample.txt -lag 1024
```

処理が終わると

```
=====
CH#      FREQ(MHz)      MAX AMP      RESIDUAL DELAY (sec)
-----
1         8209.99      0.000651      -3.26108e-06
2         8219.99      0.000443      -3.28622e-06
3         8249.99      0.000589      -3.23678e-06
4         8309.99      0.000480      -3.25464e-06
=====
```

```
COUT File is ../cout/cout0005.txt
```

のようにチャンネル毎の相関強度と遅延残差および相関処理結果ファイル (cout ファイル) 名情報が表示され、さらに相関関数のグラフ (図 7.1) も表示される。

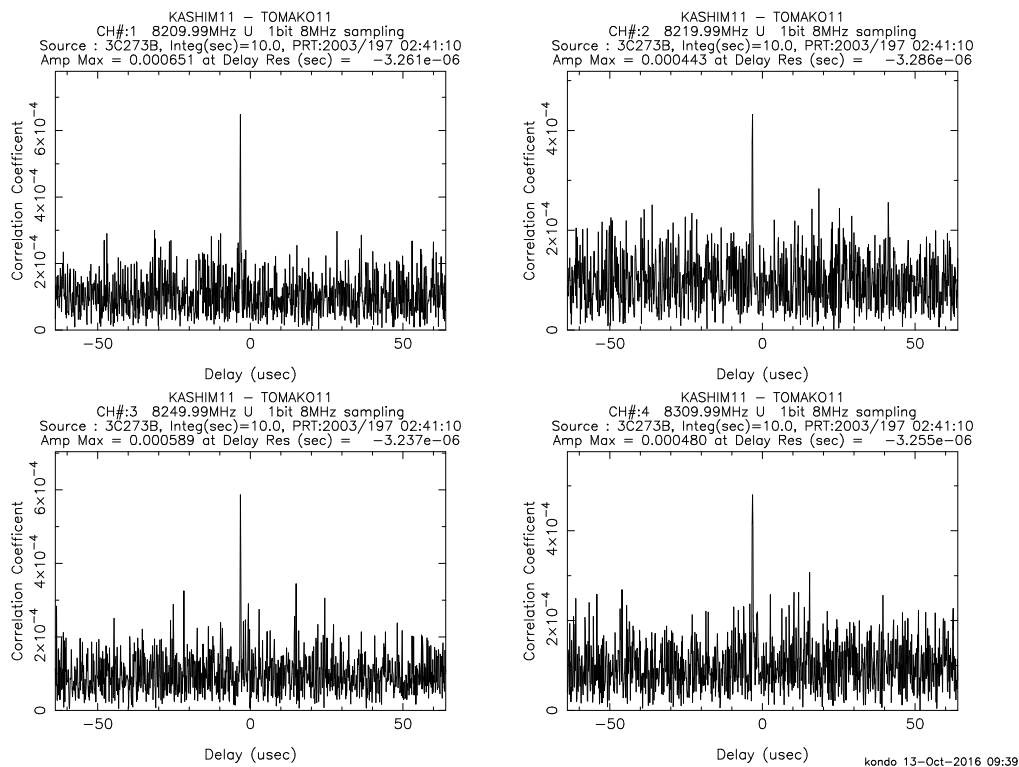


図 7.1: fx\_cor 処理終了後表示される相関関数

相関のピークが明瞭な場合はこれらの結果のみでクロックオフセットを決定できるが、sdelay を用いて粗決定サーチまで行くとより確実に判定することができる。fx\_cor 出力を使って粗決定サーチを行なうには以下のように実行する

```
sdelay ../cout/cout0005.txt
```

実行すると以下のようなサマリーとサーチ関数と呼ばれる 3D 図 (図 7.2) が表示される。

```
***** SDELAY (Ver. 2016-08-12) SUMMARY OUT PUT *****
COUT      : ../cout/cout0005.txt
X DATA   : /home/kondo/data/testspeed/Yk5data.10.dat
Y DATA   : /home/kondo/data/testspeed/Yk5data.10.dat
BASELINE  : KASHIM11 - TOMAKO11
SOURCE    : 3C273B          SAMPLING : 1 bit   8 MHz
PRT       : 2003/197 02:41:10  Tinteg(s) : 9.0
LAG SIZE  : 1024
CLOCK     : offset      5.735e-06(s)  rate   0.000e+00(s/s)
EOP       : ut1-utc 0.000000(s)
           : x-wobb 0.000000(asec)
           : y-wobb 0.000000(asec)
```

CH#	FREQUENCY (MHz)	AMP MAX	POSITION (2048x 128)	RESIDUAL Delay(usec)	Rate(ps/s)	SNR	
1	8209.99 U	1.192e-03	( 973, 65)	-3.251	-0.134	10.1	<=相 関 強 度と残差を確認
2	8219.99 U	8.613e-04	( 973, 68)	-3.270	2.482	7.3	
3	8249.99 U	1.089e-03	( 973, 65)	-3.239	-0.089	9.2	
4	8309.99 U	1.095e-03	( 973, 66)	-3.242	0.943	9.3	

Note: No amplitude correction is made.

```
===== PCAL SUMMARY =====
CH#  PCAL FREQ(kHz)  X-Amp X-Phase  Y-Amp Y-Phase
=====
```

```

1          10.00          0.109    70.6          0.251    111.9
2          10.00          0.107   -92.2          0.244   -162.7
3          10.00          0.106   136.2          0.247    84.3
4          10.00          0.104  -143.5          0.259   -43.2
*****

```

Outfile is ./sdelayout.txt

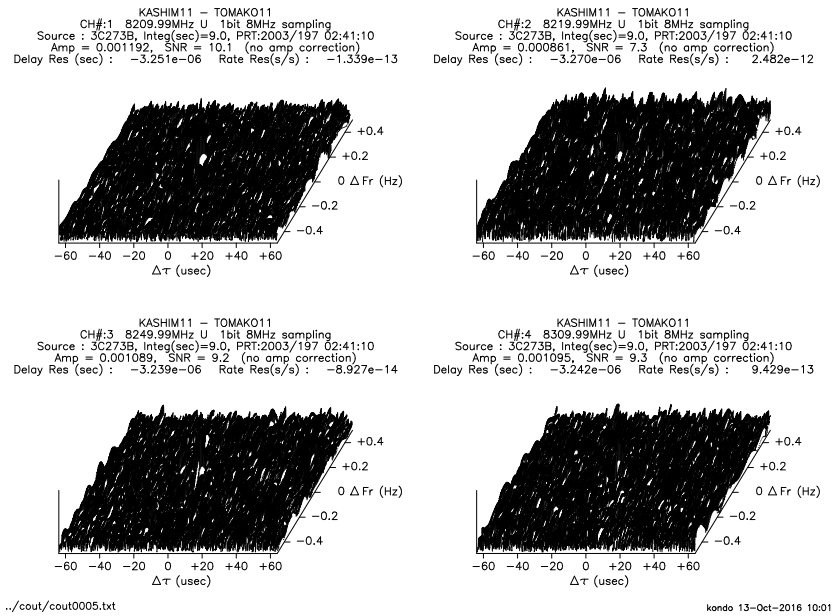


図 7.2: sdelay 処理終了後表示されるサーチ関数。ラグ数が大きいいためピークが良く分からない。こうした場合には”-tzoom”および”-tshit”オプションを使って得られた残差遅延付近を拡大すると良い。”-tzoom 20 -tshift -3.25e-6”オプションで表示したサーチ関数を図 7.3 に示す。

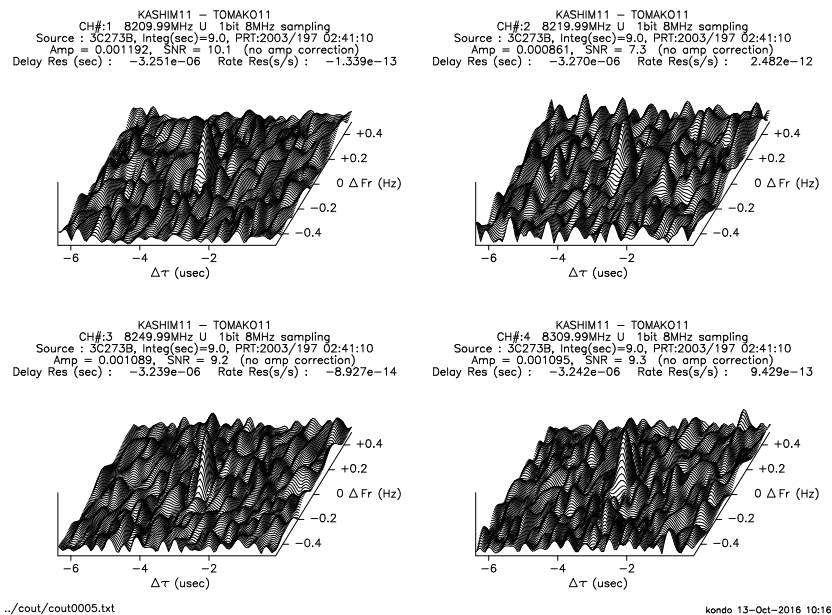


図 7.3: 図 7.2 と同じ cout ファイルを”-tzoom 20 -tshift -3.25e-6”オプションで sdelay 処理した場合のサーチ関数。



サマリー表示およびサーチ関数の 3D 表示からフリンジ検出の判断を行なう。

- POSITION で示されてピークの位置が全チャンネルでほぼ一致

していればフリンジ検出と判断しても良いが、更に

- サーチ関数で明瞭な単独ピークが出ている
- SNR が少なくとも 10 以上ある

であれば、より確実である（ここで示した処理例はチャンネルによっては SNR が 10 より若干小さいものもあるが、ピークの位置が揃っている事、および明瞭な単独ピークであることからフリンジ検出と言える）。

#### 4. クロックオフセットを決める

予測値計算時に設定したクロックオフセットを  $c_o$ 、相関処理後に得られた遅延残差を  $\Delta\tau$  とすると実際のクロックオフセット  $c_{offset}$  は次式で得られる。

$$c_{offset} = c_o + \Delta\tau$$

#### 5. クロックレートを定める

セッションの最初の方の時刻  $t_1$  で得られたクロックオフセットを  $c_1$ 、セッションの最後の方の時刻  $t_2$  で得られたクロックオフセットを  $c_2$  とするとクロックレート  $c_{rate}$  は次式で計算される。

$$c_{rate} = \frac{c_2 - c_1}{t_2 - t_1}$$

ここで  $t_1, t_2$  は秒単位で表したそれぞれの PRT（処理参照時刻）である。セッションは 0h を跨いでいない場合は時分秒のみの考慮で良いが、0h を跨いでいる場合は日まで考慮する必要がある。

#### 6. apri\_calc でのクロックパラメータの設定

apri\_calc ではクロックオフセット、クロックレートおよびクロックエポックを設定できるがそれらは以下のように設定する。

- 全スキャンで同じクロックオフセットとする場合  
クロックオフセットのみ設定し他はデフォルトもしくは 0 をセットする
- クロックレートを反映させる場合  
クロックオフセットは  $c_1$ 、レートは  $c_{rate}$ 、クロックエポックは  $t_1$ （年、通日、時、分、秒）を設定する

## 7.2 全観測（スキャン）の処理

### 1. 予測値の計算

apri\_calc を使ってフリンジサーチで得たクロックオフセット、レートおよびエポックを反映させて全スキャンの予測値を計算する。非会話型で行うには以下の例のようにオプションを指定する。

```
apri_calc /home/vlbi/sked/jd1606.skd -coffset -3.25e-6 -crate 1.0e-13 -epoch 2016/197-02:41:10
      -baseid RY -g 2 -xdir -ydir /home/vlbi/Y -apedir /home/vlbi/corrapri
```

この例ではクロックオフセットを  $-3.25\mu\text{sec}$ 、クロックレートを  $1.0 \times 10^{-13}(\text{s/s})$ 、クロックエポックを 2016 年 197 日 02 時 41 分 10 秒とし、基線 ID は"RY"、X 局データのディレクトリは"/home/vlbi/data/R"、Y 局データのディレクトリは"/home/vlbi/data/Y"、予測値ファイルの出力ディレクトリは"/home/vlbi/corrapri"としている。全スキャンで同じクロックオフセットを使う場合は"-crate"オプションおよび"-epoch"オプションを省略する。



## 2. 予測値ファイルリストの作成

予測値ファイルのあるディレクトリを”/home/vlbi/corappri”とする。以下のコマンドを実行する

```
ls -l /home/vlbi/corappri/ape*RYb.txt > apelistRY.txt
```

”ape\*RYb.txt”の部分はそのディレクトリに複数基線の予測値があった場合にファイルを抽出するフィルターであるが、全予測値を対象とする場合は

```
ls -l /home/vlbi/corappri/ape*.txt > apelistRY.txt
```

で良い。なお”apelistRY.txt”は作成するリストファイル名で任意である。

## 3. 全スキャンの相関処理

予測値ファイルリストを apelistRY.txt とした場合、以下のように実行する。

```
cor_all ./apelistRY.txt
```

ただし作業ディレクトリに予測値ファイルリストがあるとする。別のディレクトリにある場合はパスを含めて指定する。実行時のオプションについては 4.2 章を参照されたし。

cor\_all は 1 ビットサンプリングデータの処理を行う場合であるが多ビットサンプリングデータの場合は fx\_cor\_all を使用する。また K5/VSSP 以外のデータの処理を行う場合は cor\_all\_new や fx\_cor\_all\_new を使用する。

## 7.3 K5/VSSP 以外のフォーマットデータ (VDIF, Mark5B, ADS) を含む場合の相関処理

K5/VSSP データとそれ以外のフォーマットデータとの混合相関や K5/VSSP フォーマット以外のデータ同志での相関処理手順は K5/VSSP 同志の場合と同じであるが、予測値計算時にデータフォーマットを指定する必要がある。具体的には apri\_calc の実行時にオプション”-format”や”-formX”、”-formY”のオプションでデータフォーマットを指定する。詳しくは 3.1 章を参照されたし。

## 8 文書更新履歴

**2016.05.29** cor\_new, fx\_cor\_new 公開に伴う大幅改訂

**2016.12.19** sdelay に新たな機能を追加したことによる改訂。クロススペクトルグラフ出力時にクロススペクトルデータ出力も行なうように改修。

**2017.03.03** 1 章に「1.4 環境の設定」を追加

**2017.03.06** 本文に 6 章「データフォーマット変換ソフトの使い方」および付録に「各種 VLBI データフォーマットを追加」

**2019.09.07** fx\_cor, fx\_cor\_new, sdelay に新たなオプション ('-delpcal' および '-pcalbw' を追加したことによる改訂

**2019.09.15** vdifcheck, fx\_cor\_new, vdif2k5 に機能追加したことによる改訂

**2019.12.03** apri\_calc のオプション '-thread', '-threadX', '-threadY' に単 CH/スレッドのマルチスレッド VDIF データ対応機能を追加

**2019.12.21** apri\_calc で SKED ファイル使用時に、周波数と CH# の対応が局毎に異なっている場合でも適切な予測値ファイルが作成できるように改修した。また旧 Ver と互換性を保つためのオプション '-legacy' を追加した。skdchk も同様な改修を実施。これらを反映するための改訂

## A 予測値ファイルフォーマット

予測値計算ソフト “apri\_calc” によって作成される予測値ファイルのフォーマットを説明する。

### 1. 予測値ファイル構造

予測値ファイルは ‘\$’ で始まる記述子とその次の行以降の対応するパラメータで構成される。記述子はファイル中以下の表に示す順番で記述されるが、\$FORMAT1、\$FORMAT2 はデータファイルフォーマットを記述するための記述子で K5/VSSP（含む K5/VSSP32）フォーマットの場合はこれらの記述子は不要である。なお、ファイル中 ‘\*’ 以降はコメントとして扱われる

表 A.1: 予測値ファイル中で使われるパラメータ記述子

記述子	説明
\$EXPCODE	実験コード記述子
\$OBS_NUMBER	観測（スキャン）番号記述子
\$STATION1	X 局（局 1）情報記述子
\$FORMAT1	X 局（局 1）データフォーマット記述子
\$XYZ-STATION1	X 局（局 1）位置情報記述子
\$STATION2	Y 局（局 2）情報記述子
\$FORMAT2	Y 局（局 2）データフォーマット記述子
\$XYZ-STATION2	Y 局（局 2）位置情報記述子
\$BASEID	基線 ID 記述子
\$FRQ_GRP(1-4)	周波数グループ記述子
\$FREQUENCY	RF 周波数情報記述子
\$PCAL_FREQ	PCAL 周波数情報記述子
\$CLOCK	クロック情報記述子
\$SOURCE	電波源名記述子
\$RA	電波源位置赤経記述子
\$DEC	電波源位置赤緯記述子
\$EPOCH	電波源位置エポック記述子
\$GHA	電波源グリニッチ時角記述子
\$EOP	地球姿勢パラメータ記述子
\$START	スキャン開始時刻記述子
\$STOP	スキャン終了時刻記述子
\$APRIORI	予測値記述子
\$END	予測値ファイル終了記述子

## 2. 各記述子パラメータ

\$EXPCODE	実験コード記述子
<i>exp_code</i>	実験コード
\$OBS_NUMBER	観測（スキャン）番号記述子
<i>n</i>	スキャン番号
\$STATION1	X 局（局 1）情報記述子
<i>station1_name data_file</i>	X 局名 データファイル名
\$FORMAT1	X 局（局 1）データフォーマット記述子（VSSP フォーマットの場合は不要）
<i>data_format [sampling_info] [thread#]</i>	X 局データフォーマット <i>data_format</i> とサンプリング情報 <i>sampling_info</i> および スレッド番号 <i>thread#</i> （VDIF フォーマット時のみ） データフォーマットは VDIF M5B OCTAD ADS ここで VDIF – VDIF フォーマット M5B – Mark-5B フォーマット OCTAD – OCTAD フォーマット ADS – ADS フォーマット サンプリング情報はサンプリング周波数 ( <i>m</i> ) チャンネル数 ( <i>n</i> ) および AD ビット数 ( <i>k</i> ) で以下のように記述 <i>m</i> MHz <i>n</i> CH <i>k</i> bit VDIF フォーマットの場合はサンプリング情報は通常は不要であるが ヘッダー内にサンプリング情報が記述されていない場合はサンプリング情報 が必要 スレッド番号 ( <i>n</i> ) は複 <i>ch</i> /スレッドデータの処理スレッド番号（0 から始まる 番号）で以下のように記述 THREAD- <i>n</i>
\$XYZ-STATION1	X 局（局 1）位置情報記述子
<i>x y z</i>	X 座標 (m) Y 座標 (m) Z 座標 (m)
\$STATION2	Y 局（局 2）情報記述子
<i>station2_name data_file</i>	Y 局名 データファイル名
\$FORMAT2	Y 局（局 2）データフォーマット記述子（VSSP フォーマットの場合は不要）
<i>data_format [sampling_info] [thread#]</i>	Y 局データフォーマット <i>data_format</i> とサンプリング情報 <i>sampling_info</i> および スレッド番号 <i>thread#</i> （VDIF フォーマット時のみ）
\$XYZ-STATION2	Y 局（局 2）位置情報記述子
<i>x y z</i>	X 座標 (m) Y 座標 (m) Z 座標 (m)
\$BASEID	基線 ID 記述子
<i>baseline_id</i>	基線 ID（2 文字または 4 文字）
\$FRQ_GRP(1-4)	周波数グループ記述子
<i>n</i>	周波数グループ番号 (1-4) または 0 0 は全 CH 処理の場合
\$FREQUENCY	RF 周波数情報記述子
<i>rf_freq side_band [x-ch [y-ch [pol [(thx-thy)]]]]</i>	ここで <i>rf_freq</i> – RF 周波数 (Hz)、 <i>side_band</i> – サイドバンド (U L) <i>x-ch</i> – X データ CH#, <i>y-ch</i> – Y データ CH#

*pol* – 偏波情報 (例: XX, YY, RR, RL, ...) ‘--’ は情報なし

*thx* – X データスレッド# (VDIF 時有効)

*thy* – Y データスレッド# (VDIF 時有効)

単 *ch*/スレッドの複スレッドデータ処理に使用

フォーマット記述子でスレッド#を記述している場合はそちらが

優先する

\$PCAL_FREQ	PCAL 周波数情報記述子
<i>pcal_freq</i>	PCAL 周波数 (Hz)
\$CLOCK	クロック情報記述子
OFST= <i>c_offset</i>	クロックオフセット (s)。Y 局が進んでいる場合を正にとる
RATE= <i>c_rate</i>	クロックレート (s/s)
XCOF= <i>xc_offset</i>	X 局クロックと UTC との差 (s)
\$SOURCE	電波源名記述子
<i>srcnam</i>	電波源名 (8 文字)
\$RA	電波源位置赤経記述子
<i>hour minute sec</i>	赤経 (時 分 秒)
\$DEC	電波源位置赤緯記述子
<i>deg minute sec</i>	赤緯 (度 分 秒)
\$EPOCH	電波源位置エポック記述子
<i>year</i>	エポック (年)
\$GHA	電波源グリニッチ時角記述子
<i>hour minute sec</i>	時角 (時 分 秒)
\$EOP	地球姿勢パラメータ記述子
UT1-UTC= <i>ut1mutc</i>	UT1-UTC (s)
X_WOBB = <i>wobbx</i>	WobbX (arcsec)
Y_WOBB = <i>wobby</i>	WobbY (arcsec)
\$START	スキャン開始時刻記述子
<i>yyyydddhmmss</i>	スキャン開始時刻 UT (年 (4 桁)、通日 (3 桁)、時 (2 桁)、分 (2 桁)、秒 (2 桁))
\$START	スキャン終了時刻記述子
<i>yyyydddhmmss</i>	スキャン終了時刻 UT (年、通日、時、分、秒)
\$APRIORI	予測値記述子
PRT= <i>yyyydddhmmss</i>	PRT (処理参照時刻) UT (年、通日、時、分、秒)
TAU0= <i>tau</i>	予測遅延 (s)
TAU1= <i>tau1</i>	予測遅延変化率 (s/s)
TAU2= <i>tau2</i>	予測遅延 2 次変化率 (s/s <sup>2</sup> )
TAU3= <i>tau3</i>	予測遅延 3 次変化率 (s/s <sup>3</sup> )
\$END	予測値ファイル終了記述子

### 3. 予測値ファイル例

#### 例 1 . K5/VSSP 同志の場合

```

** This is Apriori file made by apri_calc Ver. 2016-09-29
**   for cor, cor_all, fx_cor, and fx_cor_all
**
** SUBNET ON: PRT is set according to each scan length
**
** Clock parameters at run are as follows,
**   Clock Offset (s) : 0.000000
**   Clock Rate (s/s) : 0.000000
**   Clock Epoch      :      0000/000 00:00:00
**
$EXPCODE      < - - 実験コード記述子
KS15002

$OBS_NUMBER   < - - 観測 (スキャン) 番号記述子
1             < - - 観測 (スキャン) 番号

$STATION1     < - - X 局 (局 1) 情報記述子
KASHIM11 ./R0020001.dat < - - X 局 (局 1) 名 データファイル名

$XYZ-STATION1 < - - X 局 (局 1) 位置情報記述子
-3997505.701700 3276878.404550 3724240.703140 < - - X 局位置 (X Y Z)(m)

$STATION2     < - - Y 局 (局 2) 情報記述子
KOGANEI ./G0020001.dat < - - Y 局 (局 1) 名 データファイル名

$XYZ-STATION2 < - - X 局 (局 1) 位置情報記述子
-3941937.479090 3368150.907990 3702235.288150 < - - Y 局位置 (X Y Z)(m)

$BASEID       < - - 基線 ID 記述子
RG            < - - 基線 ID ( 2 文字または 4 文字 )

$FRQ_GRP(1-4) < - - 周波数グループ記述子
1             < - - 周波数グループ番号 (1-4)

$FREQUENCY    < - - RF 周波数情報記述子
7864990000.0 U < - - 周波数 (Hz) サイドバンド (U|L)
7874990000.0 U
7884990000.0 U
8014990000.0 U

$PCAL_FREQ    < - - PCAL 周波数情報記述子
10000.0       < - - PCAL 周波数 (Hz)
10000.0
10000.0
10000.0

$CLOCK        < - - クロック情報記述子
OFST= 0.000000 < - - クロックオフセット
RATE= 0.000000 < - - クロックレート
XCDF= 0.000000 < - - X 局クロックと UTC との差

$SOURCE       < - - 電波源名記述子
3C345         < - - 電波源名

$RA           < - - 電波源位置赤経記述子
16 42 58.80996700 < - - 赤経 (時 分 秒)

$DEC          < - - 電波源位置赤緯記述子
39 48 36.99406000 < - - 赤緯 (度 分 秒)

$EPOCH        < - - 電波源位置エポック記述子
2000.0        < - - エポック (年)

$GHA          < - - 電波源グリニッチ時角記述子
16 3 23.584000 < - - 時角 (時 分 秒)

$EOP          < - - 地球姿勢パラメータ記述子
UT1-UTC= 0.000000
X_WOBB = 0.000000
Y_WOBB = 0.000000

$START        < - - スキャン開始時刻記述子
2015002020000 < - - YYYYDDHHMMSS

$STOP         < - - スキャン終了時刻記述子
2015002020130 < - - YYYYDDHHMMSS

```

```

$APRIORI      < - - 予測値記述子
PRT=2015002020045 < - - PRT YYYYDDHHMMSS
TAU0= -8.744597367101878e-05 < - - 予測遅延 (s)
TAU1= -1.740376052034359e-08 < - - 予測遅延変化率 (s/s)
TAU2= 7.147465473084870e-13 < - - 予測遅延 2 次変化率 (s/s^2)
TAU3= 9.254412615463208e-17 < - - 予測遅延 3 次変化率 (s/s^3)
$END          < - - 予測値ファイル終了記述子

```

## 例 2 . VDIF フォーマットデータ同志の場合

```

** This is Apriori file made by apri_calc Ver. 2016-09-29
** for cor, cor_all, fx_cor, and fx_cor_all
**
** SUBNET ON: PRT is set according to each scan length
**
** Clock parameters at run are as follows,
** Clock Offset (s) : 0.000000
** Clock Rate (s/s) : 0.000000
** Clock Epoch      : 0000/000 00:00:00
**
$EXPCODE
KS15002

$OBS_NUMBER
1

$STATION1
KASHIM11 ./R0020001.dat

$FORMAT1      < - - X 局データフォーマット記述子
VDIF          < - - VDIF フォーマット指定

$XYZ-STATION1
-3997505.701700 3276878.404550 3724240.703140

$STATION2
KOGANEI ./G0020001.dat

$FORMAT2      < - - Y 局データフォーマット記述子
VDIF          < - - VDIF フォーマット指定

$XYZ-STATION2
-3941937.479090 3368150.907990 3702235.288150

$BASEID
RG

$FRQ_GRP(1-4)
0 < - - '0' は全周波数指定を意味する

$FREQUENCY    * RFfreq U|L <pickup ch# for station1> <pickup ch# for station2>
7864990000.0 U < - - 16 c h 分の周波数を記述
7874990000.0 U
7884990000.0 U
8014990000.0 U
8114990000.0 U
8244990000.0 U
8504990000.0 U
8544990000.0 U
8564990000.0 U
8574990000.0 U
2214990000.0 U
2224990000.0 U
2234990000.0 U
2264990000.0 U
2294990000.0 U
2304990000.0 U

$PCAL_FREQ
10000.0 < - - 16 c h 分の PCAL 周波数を記述
10000.0
10000.0
10000.0

```

```

10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
$CLOCK
OFST= 0.000000
RATE= 0.000000
XCOR= 0.000000
$SOURCE
3C345
$RA
16 42 58.80996700
$DEC
39 48 36.99406000
$EPOCH
2000.0
$GHA
16 3 23.584000
$EOP
UT1-UTC= 0.000000
X_WOBB = 0.000000
Y_WOBB = 0.000000
$START
2015002020000
$STOP
2015002020130
$APRIORI
PRT=2015002020045
TAU0= -8.744597367101878e-05
TAU1= -1.740376052034359e-08
TAU2= 7.147465473084870e-13
TAU3= 9.254412615463208e-17
$END
```

### 例 3 . Mark-5B フォーマットデータ同志の場合

```

** This is Apriori file made by apri_calc Ver. 2016-09-29
**      for cor, cor_all, fx_cor, and fx_cor_all
**
** SUBNET ON:  PRT is set according to each scan length
**
** Clock parameters at run are as follows,
**      Clock Offset (s) : 0.000000
**      Clock Rate (s/s) : 0.000000
**      Clock Epoch      :      0000/000 00:00:00
**
**
$EXPCODE
KS15002

$OBS_NUMBER
1

$STATION1
KASHIM11 ./R0020001.dat

$FORMAT1
M5B 16MHz 16CH 1bit  < - - Mark-5B フォーマット指定とサンプリング情報

$XYZ-STATION1
-3997505.701700 3276878.404550 3724240.703140

```



```
$STATION2
KOGANEI ./G0020001.dat

$FORMAT2
M5B 16MHz 16CH 1bit < - - Mark-5B フォーマット指定とサンプリング情報

$XYZ-STATION2
-3941937.479090 3368150.907990 3702235.288150

$BASEID
RG

$FRQ_GRP(1-4)
0

$FREQUENCY
7864990000.0 U
7874990000.0 U
7884990000.0 U
8014990000.0 U
8114990000.0 U
8244990000.0 U
8504990000.0 U
8544990000.0 U
8564990000.0 U
8574990000.0 U
2214990000.0 U
2224990000.0 U
2234990000.0 U
2264990000.0 U
2294990000.0 U
2304990000.0 U

$PCAL_FREQ
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0
10000.0

$CLOCK
OFST= 0.000000
RATE= 0.000000
XCOF= 0.000000

$SOURCE
3C345

$RA
16 42 58.80996700

$DEC
39 48 36.99406000

$EPOCH
2000.0

$GHA
16 3 23.584000

$EOP
UT1-UTC= 0.000000
X_WOBB = 0.000000
Y_WOBB = 0.000000

$START
2015002020000

$STOP
2015002020130

$APRIORI
PRT=2015002020045
TAU0= -8.744597367101878e-05
TAU1= -1.740376052034359e-08
TAU2= 7.147465473084870e-13
TAU3= 9.254412615463208e-17
```

\$END

#### 例4 . VSSP フォーマットデータと VDIF フォーマットデータ間の相関処理の場合

```

** This is Apriori file made by apri_calc Ver. 2016-09-29
**      for cor, cor_all, fx_cor, and fx_cor_all
**
** SUBNET ON:  PRT is set according to each scan length
**
** Clock parameters at run are as follows,
**   Clock Offset (s) : 0.000000
**   Clock Rate (s/s) : 0.000000
**   Clock Epoch      :      0000/000 00:00:00
**
$EXPCODE
KS15002

$OBS_NUMBER
1

$STATION1
KASHIM11 ./R0020001.dat

$XYZ-STATION1
-3997505.701700 3276878.404550 3724240.703140

$STATION2
KOGANEI ./G0020001.dat

$FORMAT2
VDIF

$XYZ-STATION2
-3941937.479090 3368150.907990 3702235.288150

$BASEID
RG

$FRQ_GRP(1-4)
3

$FREQUENCY
8564990000.0 U 1 9    < - - RF 周波数 サイドバンド X データ CH# Y データ CH#
8574990000.0 U 2 10
2214990000.0 U 3 11
2224990000.0 U 4 12

$PCAL_FREQ
10000.0
10000.0
10000.0
10000.0

$CLOCK
OFST= 0.000000
RATE= 0.000000
XCOF= 0.000000

$SOURCE
3C345

$RA
16 42 58.80996700

$DEC
39 48 36.99406000

$EPOCH
2000.0

$GHA
16 3 23.584000

$EOP
UT1-UTC= 0.000000
X_WOBB = 0.000000
Y_WOBB = 0.000000

$START
2015002020000

```

```

$STOP
2015002020130

$APRIORI
PRT=2015002020045
TAU0= -8.744597367101878e-05
TAU1= -1.740376052034359e-08
TAU2= 7.147465473084870e-13
TAU3= 9.254412615463208e-17

$END

```

#### 例5 . VGOS データ (VDIF フォーマット) 間の相関処理の場合 – 偏波情報記述 –

```

** This is Apriori file made by apri_calc Ver. 2019-06-16
**   for cor, cor_all, fx_cor, and fx_cor_all
**
** (X clock offset wrt UTC is not reflected to a-priori values)
**
** SUBNET ON: PRT is set according to each scan length
**
** Clock parameters at run are as follows,
**   Clock Offset (s) : -4.212500000000000e-04
**   Clock Rate (s/s) : 0.000000
**   Clock Epoch      : 0000/000 00:00:00
**
$EXPCODE
v9715a

$OBS_NUMBER
80

$STATION1
SESHAN13 D:\data\CheckAtSHA0\v9715a\sv\v9715asv_no0080_1.10sec.vdif

$FORMAT1
VDIF 64MHz 8CH 2bit

$XYZ-STATION1
-2831686.993000 4675733.639000 3275327.641000

$STATION2
TIANMA13 D:\data\CheckAtSHA0\v9715a\tv\v9715atv_no0080_1.10sec.vdif

$FORMAT2
VDIF 64MHz 8CH 2bit

$XYZ-STATION2
-2826837.140000 4679223.144000 3274511.526000

$BASEID
SVTV

$FRQ_GRP(1-4)
1 2

$FREQUENCY
3480400000.0 L 1 1 XY    <== 偏波情報 XY
3448400000.0 L 2 2 XY
3384400000.0 L 3 3 XY
3320400000.0 L 4 4 XY
3224400000.0 L 5 5 XY
3096400000.0 L 6 6 XY
3064400000.0 L 7 7 XY
3032400000.0 L 8 8 XY

$PCAL_FREQ
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

$CLOCK
OFST= 0.000000

```

```

RATE= 0.000000
XCOF= 0.000000

$SOURCE
3C273B

$RA
12 29 6.69972950

$DEC
2 3 8.59828500

$EPOCH
2000.0

$GHA
15 59 50.752000

$EOP
UT1-UTC= 0.000000
X_WOBB = 0.000000
Y_WOBB = 0.000000

$START
2019196085700

$STOP
2019196085730

$APRIORI
PRT=2019196085715
TAU0= -4.230495720005300e-04
TAU1= -1.445886059562836e-09
TAU2= 1.006184911623976e-14
TAU3= 7.823024635496742e-18

$END

```

#### 4. 特殊処理予測値ファイル例

テキストエディタで\$FREQUENCY 記述子のパラメータを修正することにより処理チャンネル数を変更したり X 局と Y 局とのチャンネル対応を変更することが出来る。なお処理チャンネル数を変更した場合は対応する \$PCAL\_FREQ 記述子のパラメータも変更すること。

例 1 . 16CH データを 6CH 処理に変更し対応する Y 局 CH#も変更した例

```

$FREQUENCY
7864990000.0 U 1 6
7874990000.0 U 2 5
7884990000.0 U 3 4
8014990000.0 U 4 3
8114990000.0 U 5 2
8244990000.0 U 6 1

$PCAL_FREQ
10000.0 < - - 6 c h 分の PCAL 周波数を記述
10000.0
10000.0
10000.0
10000.0
10000.0

```

## B K5 ソフトウェア相関器出力フォーマット (FORMAT 7 形式)

以前の FORMAT 7 (2014-09-11 改訂) との相違点: Y 局の AD ビット数が X 局と異なる場合 AD ビット数のフィールドにスペースを空けて Y 局の AD ビット数を記述。

表 B.1: K5 ソフトウェア相関器出力フォーマット

行番号	項目
1	“#FORMAT7” コメント — フォーマット形式記述 (固定) + コメント コメントの中身は処理プログラム名とフリンジ回転パラメータ
	“fx_cor” でフィルタリング処理を行なった場合は以下の例のような A1~A5 が挿入される
A1	# BPF parameters (BPF を M 回設定した例)
A2(1)	# flow(MHz)-fhigh(MHz) factor : 1.250000-1.450000 1.000000 (最初の BPF パラメータ)
	以降 BPF パラメーターを設定数だけ繰り返す
A2(M)	# flow(MHz)-fhigh(MHz) factor : 1.650000-1.850000 1.000000 (最後 (M 番目) の BPF パラメータ)
A3	# Adopted frequency resolution (MHz) = 0.040000
A4	# Output lag size = 2048
A5	# FFT size for processing = 2048
	“fx_cor” で PCAL 除去処理を行なった場合は以下の例のような B1~B5 が挿入される
B1	# PCAL rejection parameters (CH 数が N の場合)
B2(1)	# CH#01 start_freq(MHz) = 0.40 interval(MHz) = 5.0 (CH#01 の PCAL 除去パラメータ)
	以降 PCAL 除去パラメーターを CH 数だけ繰り返す
B2(N)	# CH#N start_freq(MHz) = 2.40 interval(MHz) = 5.0 (CH#N の PCAL 除去パラメータ)
B3	# Bandwidth to reject (MHz) = 0.50
	以降の行番号はフィルタリングパラメーターおよび PCAL 除去処理がない場合の行番号
2	ホスト名 — 処理 PC ホスト名
3	実験コード
4	観測 (スキャン) 番号 (1 から始まる)
5	基線 I D
6	相関処理日時 年 通日 時 分 秒 月 日
7	X 局名
8	X 局位置 (x,y,z) (m)
9	X 局データファイル名
10	Y 局名
11	Y 局位置 (x,y,z) (m)
12	Y 局データファイル名
13	電波星名
14	電波星赤経 時 分 秒
15	電波星赤緯 度 分 秒
16	電波星位置エポック
17	処理参照時刻 (P R T) でのグリニッチ視恒星時 時 分 秒
18	観測開始時刻 (年 通日 時 分 秒)

19	観測終了時刻 (年 通日 時 分 秒)
20	P R T (処理参照時刻) (年 通日 時 分 秒)
21	P R Tにおける $\tau$ (遅延時間) の予測値 (sec)
22	P R Tにおける $\dot{\tau}$ (遅延時間変化率) の予測値 (s/s)
23	P R Tにおける $\ddot{\tau}$ の予測値 (s/s <sup>2</sup> )
24	P R Tにおける $d\ddot{\tau}/dt$ の予測値 (s/s <sup>3</sup> )
25	クロックオフセット (sec), X局時計のUTCに対する同期誤差 (sec) (Y局時計が進んでいる場合が正) (X局時計のUTCに対する進みを正)
26	クロックレート (s/s)
27	UT1-UTC (sec), Wob X (arcsec), Wob Y (arcsec) — 地球姿勢パラメータ
28	CH 数 [N] (1 または 4)
29	CH-1 R F 周波数 (Hz), P C A L 周波数 (Hz), サイドバンド情報 (1:USB, 0:LSB) [, X-CH#, Y-CH#[, 偏波]] ...
29+N	CH-N R F 周波数 (Hz), P C A L 周波数 (Hz), サイドバンド情報 [, X-CH#, Y-CH#[, 偏波]]
29+N	サンプリング周波数 (Hz)
30+N	X 局 AD ビット数 (1, 2, 4, または 8) Y 局 AD ビット数 (同じ場合は省略可)
31+N	単位積分時間 [PP] (sec)
32+N	全積分時間 (sec)
33+N	ラグ数 [L]
34+N	P P 数 [K]
35+N	“PP# 1” — PP#1 結果出力の開始
36+N	ラグ# C H # 相関データ実部 相関データ虚部 以降 N×L 行繰り返す
36+N(1+L)	“VALIDITY FLAG, FRACTIONAL BIT and FRINGE PHASE (APRIORI)” — 固定文字列
37+N(1+L)	vflag dtime ibit fbit frphase1 [frphase2 frphase3 frphase4] ここで vflag – データ正当性フラグ (1: 正常, 0: 1 PP 前の処理でエラーの可能性あり) dtime – BOPP (PP の先頭) での時刻 (sec: 0 時からの経過秒) ibit – BOPP での遅延 (サンプリング周期単位) の整数値 fbit – BOPP での遅延 (サンプリング周期単位) の少数部 frphase1 – ch1 のフリンジ位相の予測値 (度) (BOPP での値) frphase2 – ch2 のフリンジ位相の予測値 (度) (BOPP での値) frphase3 – ch3 のフリンジ位相の予測値 (度) (BOPP での値) frphase4 – ch4 のフリンジ位相の予測値 (度) (BOPP での値)
38+N(1+L)	“X-PCAL” — X局 P C A L 情報の開始を示す固定文字列
39+N(1+L)	m ns PCALR PCALI AMP PHASE ここで m – チャンネル# ns – PCAL 検出に使ったサンプル数 PCALR – P C A L 検出実部 PCALI – P C A L 検出虚部

	AMP – P C A L 強度
	PHASE – P C A L 位相 ( 度 )
以降 N ( c h 数 ) 回繰り返し	
39+N(2+L)	“Y-PCAL” — Y局 P C A L 情報の開始を示す固定文字列
40+N(2+L)	m ns PCALR PCALI AMP PHASE — Y局 P C A L データ
以降 N ( c h 数 ) 回繰り返し	
40+N(3+L)	“PP# 2” — PP#2 結果出力の開始
以降 K ( 全 P P 数 ) 回繰り返し	
...	

## C KSP 型相関データファイルフォーマット

K5 ソフトウェア相関器出力 (COUT 型ファイルデータ) を KOMB 処理する際は、最初に COUT 型ファイルは KSP 型相関データファイルに変換される。その後、変換された KSP 型相関データファイルを入力としてバンド幅合成処理が行われる。KSP 型相関データファイルではラグ数が 32 の固定値であるため、例えばソフト相関処理時のラグ数が 32 より少ない場合は、予測値との整合性を保ちつつゼロデータの追加を行い 32 ラグデータを生成している。またラグ数が 32 より多い場合は、32 ラグ分の相関データを抽出している。抽出するデータはデフォルトでは残差遅延 0 を中心としているが、オペレータにより任意のラグを中心とした 32 ラグ分のデータの抽出が可能である。この場合、予測値には抜き出したデータと整合性を保つよう修正が加えられる。

ところで、RF ダイレクトサンプリング実験の相関データのように広範なラグの相関データをそのまま処理する必要性が生じてきた。そのため、任意のラグ数データをそのまま KOMB で扱えるように、KSP 相関データフォーマットの拡張を定義した。

### C.1 KSP 相関データフォーマットの拡張

拡張フォーマットのヘッダー部 (HD) (512 バイト) はオリジナルとほぼ同じ形式であるが、CRSMODE に新たに拡張フォーマットであることを示すために “F” を定義した (FULL の意味)。フォーマットの拡張とともに相関カウント値も 24 ビットから 32 ビットに変更した。更にラグ数の項目 (LAG) と AD 分解能の項目 (ADBIT) を追加した。

積分単位 (PP と呼ぶ) 毎の相関データ (CD) は、ユニット毎の相関データ (UD) から構成される点もオリジナルと同じであるが、UD はオリジナルの 256 バイトサイズからラグの大きさにより 256 バイト単位で増えて行くフォーマットとした。そのため、項目は大幅に変更している。

HD	CD	CD	.....
512 バイト			

図 C.1: 相関ファイル内のレコード構成。HD:ヘッダー部、CD:積分単位 (PP) 毎の相関積分データ。

CD(PP 毎の相関データ)				
UD (1unit)	UD (2unit)	UD (3unit)	.....	UD (K unit)

図 C.2: 積分単位 (PP) 毎の相関積分データ (CD) 内のレコード構成。UD:ユニット (チャンネル) 毎の相関積分データ。

UD(ユニット毎の相関データ)				
UD#0	UD#1	UD#2	.....	UD#N
256 バイト	256 バイト	256 バイト		256 バイト

図 C.3: ユニット毎の相関積分データ (UD) 内のレコード構成。UD#0 には時刻情報等が入っており、UD#1 以降に相関データが 32 ラグ毎に入っている。例えばラグ数 64 のデータは UD#2 で終わる。ラグ数が 1024 の場合は UD#32 まで続くことになる。



## C.2 関連生データファイルデータフォーマット：ヘッダーレコード (HD)

(cout2ksp および komb の Ver.2016-08-5 以降の変更箇所は赤文字で表記)

表 C.1: K S P 関連器出力データフォーマット：ヘッダーレコード (HD)(512 バイト)

シンボル	バイト数	バイト位置	タイプ	説明
EXCODE	10	1	A10	実験コード (10 文字)
NOBS	2	11	I*2	観測番号 (注: H P - U X では INTEGER は 4 バイトになる)
LFILE	6	13	A6	関連器出力ファイル名 (6 文字)
LBASE	2	19	A2	基線 I D (2 文字)
NPP	2	21	I*2	P P 数
NPPSEC	2	23	I*2	P P 時間
				FMTFLAG “KSP” と “K4” は P P 時間単位は sec。
				FMTFLAG “KSP1” は P P 時間単位は 10msec。
NKOMB	2	25	I*2	バンド幅合成処理回数 (K O M B が処理する度に + 1 される)
KRDATE	8	27	I*2	DIM(4) 相関処理日時 (年、通日、時、分)
KBFILE	6	35	A6	K O M B 処理結果ファイル名 (K O M B がセットする)
SRCNAM	8	41	A8	電波源名 (8 文字)
SRCRA	4	49	I*2	DIM(2) 電波源の赤経 ( $\alpha$ ) (時、分) J 2 0 0 0 年系
	8	53	R*8	電波源の赤経 ( $\alpha$ ) (秒部分) J 2 0 0 0 年系
SRCDEC	4	61	I*2	DIM(2) 電波源の赤緯 ( $\delta$ ) (度、分) J 2 0 0 0 年系
	8	65	R*8	電波源の赤緯 ( $\delta$ ) (秒部分) J 2 0 0 0 年系
IPRT	10	73	I*2	DIM(5) P R T (処理参照時刻: 観測のほぼ中央の時刻) (年、通日、時、分、秒)
STATX	8	83	A8	X 局名 (8 文字)
STATY	8	91	A8	Y 局名 (8 文字)
X_XYZ	24	99	R*8	DIM(3) X 局位置 (X, Y, Z) (m)
Y_XYZ	24	123	R*8	DIM(3) Y 局位置 (X, Y, Z) (m)
OSTART	10	147	I*2	DIM(5) 観測開始時刻 (年、通日、時、分、秒)
OSTOP	10	157	I*2	DIM(5) 観測終了時刻 (年、通日、時、分、秒)
SRCGHA	4	167	I*2	DIM(2) P R T での電波源のグリニッチ時角 (時、分)
	8	171	R*8	P R T での電波源のグリニッチ時角 (秒部分)
TSAMPL	4	179	R*4	サンプリング周期 (sec)
VBW	4	183	R*4	ビデオ帯域幅 (Hz)
NCH	2	187	I*2	相関処理時のチャンネル数
ACLKO	4	189	R*4	クロックオフセットのアプリオリ値 (sec) P R T における X 局と Y 局の時刻同期誤差。Y 局が進んでいる場合を正にとる。
ACLKR	4	193	R*4	P R T における X 局と Y 局のクロックレート差 (s/s)
DLYINX	4	197	R*4	X バンドの X 局と Y 局の局内遅延差 (sec)
DLYINS	4	201	R*4	S バンドの X 局と Y 局の局内遅延差 (sec)
AXCLKE	4	205	R*4	P R T における X 局のクロックと U T C の同期誤差のアプリオリ値 (sec)
PI	8	209	R*8	$\pi$

C	8	217	R*8	光速度 (m/s)
FRQTAB	128	225	R*8	DIM(16) R F 周波数テーブル (Hz) 正の値: U S B、負の値: L S B
PCALF	64	353	R*4	DIM(16) P C A L (位相校正信号) 周波数テーブル (Hz)
APTAU	32	417	R*8	DIM(4) 相関処理時のアプリオリ値 $\tau(\text{sec}), \dot{\tau}(\text{s/s}), \ddot{\tau}(\text{s/s}^2), \dddot{\tau}(\text{s/s}^3)$
SRCH	2	449	I*2	フリンジサーチモード時の共通チャンネル番号 (1 ~ 16)
CMODE	2	451	A2	相関器モード “NO”: NORMAL (通常) モード、“SE”: フリンジサーチモード
UINT	2	453	I*2	フリンジサーチモード (ラグ連結) 時のユニット間ラグ数 (デフォルト値は 30)
CUNIT	2	455	I*2	フリンジサーチモード時のラグ 0 を含むユニット番号
CRLDBL	8	457	R*8	制御計算機からセット可能な倍精度実数 (未使用)
CRLNG	4	465	I*4	制御計算機からセット可能な 4 バイト整数 (未使用)
CRLSHT	2	469	I*2	制御計算機からセット可能な 2 バイト整数 (未使用)
FRGMOD	2	471	A2	フリンジ回転モード “CO”: 連続、“EV”: P P 毎初期設定
CRSMODE	1	473	A1	相関積分および P C A L カウンター出力モードフラグ 積分カウンターを 28 ビットとして “U”: 上位 24 ビット出力、“L”: 下位 24 ビット出力 “H”: カウンターを 32 ビットと仮定して上位 24 ビット出力 “F”: 32 ビットフルに出力
VER	8	474	A8	相関器 ROM バージョン CRSMODE “F” の場合は “K5-WIDE ”
—	1	482	—	未使用
JXOFST	4	483	I*4	X 局出力 I/F 装置のオフセット遅延 (ビット単位)
JYOFST	4	487	I*4	Y 局出力 I/F 装置のオフセット遅延 (ビット単位)
LAG	4	491	I*4	CRSMODE “F” の場合はここにラグ数を記述 それ以外の場合は未使用 (または 32 固定値)
ADBIT	4	495	I*4	AD 分解能
ADBITY	4	499	I*4	Y 局 AD 分解能 (CORTYPE がセットされている場合)
CORTYPE	2	503	A2	相関器タイプ “Xf” または “Fx”
—	4	505	—	未使用
FMTFLAG	4	509	A4	フォーマット識別フラグ。 “KSP ”, “K4 ”, “KSP1”, “KSP2” “KSP ” と “K4 ” は P P の単位は sec。 “KSP1” は 10msec、“KSP2” は 1msec。

## C.3 関連生データファイルデータフォーマット：ユニット毎の関連積分データ (UD)

表 C.2: K S P 関連器出力データフォーマット:ユニットレコード  
(UD)(256 バイト)

シンボル	バイト数	バイト位置	タイプ	説明
RMKS	2	1	2BYTE	リマークス。関連器がセットする 2 バイトデータ バイト# 1 : KSEL ( フリンジローテーション時の K の値 ) バイト# 2 : BIT#(LSB=0) 7-3: チャンネル番号 (1-16) 2: KOMB がセットする消去フラグ 1:消去 1-0: 未使用
COFLG	1	3	BYTE	関連処理フラグ BIT#(LSB=0) 7-6: フリンジ符号 10 ... 負の回転 00 ... PP の途中で符号反転 01 ... 正の回転 5: フリンジローテータモード 1 ... ハードで行う 0 ... ソフトで行う 4: フリンジ回転周波数 1 ... ビデオ帯域の中心周波数 0 ... ベースバンド周波数 3: 部分ビット補正 1 ... ハードで行う 0 ... ソフトで行う 2: PP パラメータ更新フラグ 1 ... パラメータ更新あり 0 ... パラメータ更新なし 1-0: 未使用
TWESTS	1	4	BYTE	積分処理ステータス BIT#(LSB=0) 7: AVL 積分有効フラグ 1 ... 積分有効 0 ... 積分無効 6-0: 未使用
CROSP	192	5	I*3	DIM(64) 関連積分データ (PP 毎リセット) ( 実数部 3 バイト× 32 ラグ、虚数部 3 バイト× 32 ラグ ) 積分カウンタは 28 ビットカウンタ ヘッダー部の CRSMODE に応じて “L”:下位 24 ビット、“U”:上位 24 ビットが出力される “H”:32 ビットカウンタの上位 24 ビットが出力される
COUNTP	8	197	I*4	DIM(2) 関連積分に関与したビット数のカウント値 ( 実数部 4 バイト、虚数部 4 バイト )
PCALD	12	205	I*3	DIM(4) 位相校正 (PCAL) 信号検出カウント値 (PP 毎リセット)

				X局 PCAL データ実数部 (3 バイト) X局 PCAL データ虚数部 (3 バイト) Y局 PCAL データ実数部 (3 バイト) Y局 PCAL データ虚数部 (3 バイト) 信号検出カウンターは 28 ビットカウンター ヘッダー部の CRSMODE に応じて “L”:下位 24 ビット、“U”:上位 24 ビットが出力される “H”:32 ビットカウンターの上位 24 ビットが出力される
TIMX	7	217	14 × 4bits	X局時刻ラベル: YYDDDDHHMMSSmmm ( 1 桁 4 ビットの 16 進数で表現)
TIMY	7	224	14 × 4bits	Y局時刻ラベル: YYDDDDHHMMSSmmm ( 1 桁 4 ビットの 16 進数で表現)
TMDIFF	4	231	I*4	X、Y局の時系列のオフセット値 (ビット単位) Y局が進んでいるときを正にとる
FRADD	4	235	32bits	PP 最後でのフリンジローテータアドレス (32 ビット) フリンジ回転をハードでやらない場合は PP での フリンジ位相の予測 (ローテータアドレス) 値
IFBIT	2	239	I*2	部分ビット値。PP での遅延予測値 (ビット単位) の 小数部。-32768 ~ 32767 が -0.5 ~ +0.5 に対応。
MODE	1	241	BYTE	各種モード BIT#(LSB=0) 7-2: 未使用 1: 2 / 1 ビットモード 1 ... 2 ビット相関 ( K 4 のみ ) 0 ... 1 ビット相関 2: 2 ビット相関時のウェイトモード 1 ... Weight モード 0 ... Binary モード
IPP	2	242	I*2	PP 番号
—	13	244	—	未使用

#### C.4 相関生データファイルデータフォーマット (CRSMODE=“F” の場合のみ): ユニット毎の相関積分データ (UD)

以下に記述するのは CRSMODE=“F” の場合のフォーマットであり、それ以外の場合は従来と同じフォーマットである。

表 C.3: K S P 相関器出力データフォーマット (UD#0)(最初の 256 バイト)

シンボル	バイト数	バイト位置	タイプ	説明
RMKS	2	1	2BYTE	リマークス。相関器がセットする 2 バイトデータ バイト# 1: KSEL (フリンジローテーション時の K の値) バイト# 2: BIT#(LSB=0) 7-3: チャンネル番号 (1-16) 2: KOMB がセットする消去フラグ 1:消去 1-0: 未使用
COFLG	1	3	BYTE	相関処理フラグ BIT#(LSB=0) 7-6: フリンジ符号 10 ... 負の回転 00 ... PP の途中で符号反転 01 ... 正の回転 5: フリンジローテータモード 1 ... ハードで行う 0 ... ソフトで行う 4: フリンジ回転周波数 1 ... ビデオ帯域の中心周波数 0 ... ベースバンド周波数 3: 部分ビット補正 1 ... ハードで行う 0 ... ソフトで行う 2: PP パラメータ更新フラグ 1 ... パラメータ更新あり 0 ... パラメータ更新なし 1-0: 未使用
TWESTS	1	4	BYTE	積分処理ステータス BIT#(LSB=0) 7: AVL 積分有効フラグ 1 ... 積分有効 0 ... 積分無効 6-0: 未使用
*** これ以降の項目は大幅に変更されている ***				
TIMX	7	5	14 x 4bits	X 局時刻ラベル: YYDDDDHHMMSSmmmm (1 桁 4 ビットの 16 進数で表現)
TIMY	7	12	14 x 4bits	Y 局時刻ラベル: YYDDDDHHMMSSmmmm (1 桁 4 ビットの 16 進数で表現)

TMDIFF	4	19	I*4	X、Y局の時系列のオフセット値（ビット単位） Y局が進んでいるときを正にとる
FRADD	4	23	32bits	PP 最後でのフリンジローテータアドレス（32 ビット） フリンジ回転をハードでやらない場合は PP での フリンジ位相の予測（ローテータアドレス）値
IFBIT	2	27	I*2	部分ビット値。PP での遅延予測値（ビット単位）の 小数部。-32768 ~ 32767 が -0.5 ~ +0.5 に対応。
MODE	1	29	BYTE	各種モード（未使用） BIT#(LSB=0) 7-2: 未使用 1: 2 / 1 ビットモード 1 ... 2 ビット相関（K 4 のみ） 0 ... 1 ビット相関 2: 2 ビット相関時のウェイトモード 1 ... Weight モード 0 ... Binary モード
IPP	2	30	I*2	PP 番号
PCALD	16	32	I*4	DIM(4) 位相校正 (PCAL) 信号検出カウント値 X局 PCAL データ実数部（4 バイト） X局 PCAL データ虚数部（4 バイト） Y局 PCAL データ実数部（4 バイト） Y局 PCAL データ虚数部（4 バイト）
COUNTP	8	48	I*4	DIM(2) 相関積分に関与したビット数のカウント値 （実数部 4 バイト、虚数部 4 バイト）
—	—	56	—	256 バイトの最後まで未使用

表 C.4: K S P 相関器出力データフォーマット (UD#1)(256 バイト)

シンボル	バイト数	バイト位置	タイプ	説明
CROSP	4	1	I*4	ラグ# 1 相関積分データ実数部
	4	5	I*4	ラグ# 2 相関積分データ実数部
				...
	4	125	I*4	ラグ# 32 相関積分データ実数部
	4	129	I*4	ラグ# 1 相関積分データ虚数部
	4	133	I*4	ラグ# 2 相関積分データ虚数部
				...
	4	253	I*4	ラグ# 32 相関積分データ虚数部

ラグ数が 32 より大きい場合は、以下のように最大ラグを満たすまでレコードが 256 バイト単位（32 ラグ単位）で追加される

表 C.5: K S P相関器出力データフォーマット (UD#2)(256 バイト)

シンボル	バイト数	バイト位置	タイプ	説明
CROSP	4	1	I*4	ラグ# 33 相関積分データ実数部
	4	5	I*4	ラグ# 34 相関積分データ実数部
				...
	4	125	I*4	ラグ# 64 相関積分データ実数部
	4	129	I*4	ラグ# 33 相関積分データ虚数部
	4	133	I*4	ラグ# 32 相関積分データ虚数部
				...
	4	253	I*4	ラグ# 64 相関積分データ虚数部

表 C.6: K S P相関器出力データフォーマット (UD#N)(256 バイト)

シンボル	バイト数	バイト位置	タイプ	説明
CROSP	4	1	I*4	ラグ# $32 \times (N - 1) + 1$ 相関積分データ実数部
	4	5	I*4	虚数部
	4	9	I*4	ラグ# $32 \times (N - 1) + 2$ 相関積分データ実数部
	4	13	I*4	虚数部
				...
	4	249	I*4	ラグ# $32 \times N64$ 相関積分データ実数部
	4	253	I*4	虚数部

## D 各種 VLBI 観測データフォーマット

### D.1 データフォーマットの種類

VLBI データ構造はヘッダーおよびデータ部のフレームデータから構成されるもの (Type-1) とフレームを構成しないサンプリング時系列データを並べたのみのもの (Type-2) の 2 種類に分けられる。前者は更にデータの一部がヘッダー部に置き換えられたもの (その部分のサンプリングデータは欠損する) (Type-1A) とヘッダー部がデータ間に挿入 (サンプリングデータの欠損は生じない) (Type-1B) の 2 つに分かれる (図 D.1)。VLBI 技術が実用化された当時は Type-1A (Mark-III, Mark-IV, K-3) フォーマットのみであったが、現在は Type-1B (VLBA, Mark-5B, VDIF, K5/VSSP, VSSP32/64, ADS の DBBC モード) フォーマットが主流であり、Type-2 (OCTAD, ADS 生データモード) も実観測に使用されている。

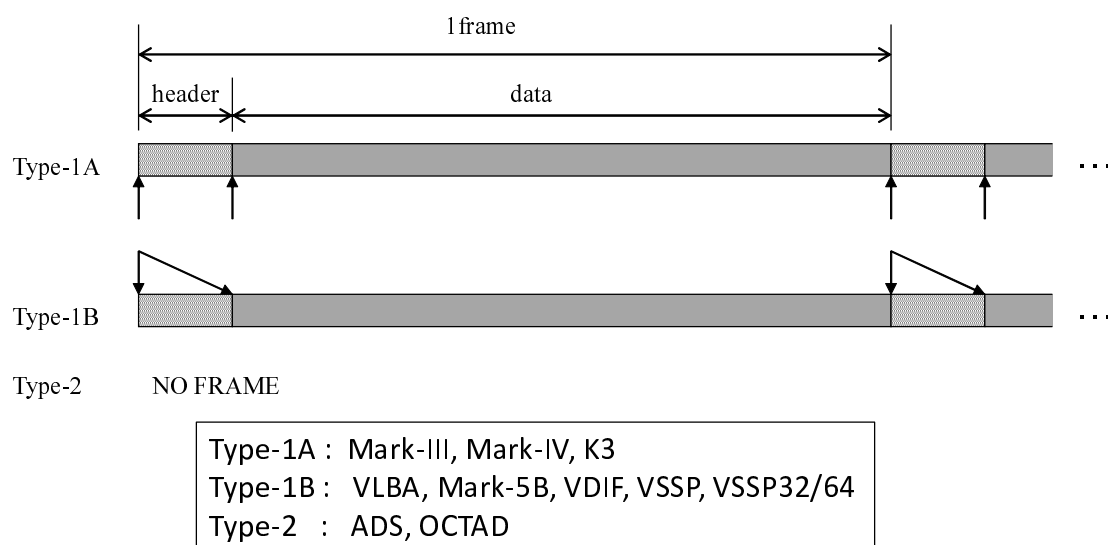


図 D.1: VLBI データ構造の種類

表 D.1 に各フォーマットのフレーム長およびヘッダー部の諸元を示す。

表 D.1: 各フォーマットの比較

	フレーム長 (バイト)	ヘッダー部 (バイト)	データ部 (バイト)
Mark-III/IV, K3	2500	20	2480
VLBA	2520	20	2500
Mark-5B	10016	16	10000
VDIF	可変	32	可変
VSSP	1 frame/sec	8	可変
VSSP32/64	1 frame/sec	32	可変

これらのフォーマットの内、NICT が制定した VSSP フォーマット (単に VSSP や K5/VSSP フォーマットと称した場合は VSSP32, VSSP64 フォーマットを含む)、現在国際的な主流である Mark5B フォーマット、今後主流となるであろう VDIF フォーマットについて説明する。



## D.2 K5/VSSP, VSSP32 および VSSP64 フォーマット

### D.2.1 データ構造

K5/VSSP, K5/VSSP32 および K5/VSSP64 (以降、単に VSSP, VSSP32, VSSP64 と記述する) サンプラーからの出力データは 1 秒毎のフレームデータで構成されており、フレームデータは同期パターンを含むヘッダー部 (HD) とサンプリングデータ部 (SD) からなる (図 D.2)。VSSP データのヘッダーは 8 バイト (64 ビット) で構成され同期パターン、時刻データ、サンプリングデータの A/D ビット数およびサンプリング周波数に関する情報が含まれる。VSSP32 のヘッダー部は通常 32 バイト (256 ビット) で構成され最初の 8 バイトは VSSP のヘッダー部と同じ情報 (但しシンクパターンの一部が異なる) である。VSSP32 のヘッダー部には更に年、通日の情報等が追加されている。VSSP64 のヘッダー部はサンプラーが VSSP32 モードで動作<sup>3</sup>するときは VSSP32 と全く同じである (VSSP64 では 2ch モードが可能であるが、その際は拡張フォーマットを使用する)。VSSP64 モードで動作する際はほとんどが VSSP32 フォーマットと同じだが第 2 同期パターンと EF (エラーフラグ) の部分が異なっており、VSSP64 モードでは第 2 同期パターンが 0x8D であり、EF はチャンネル数情報の補助フラグとして使われ 2 ch モードではここが 1 にセットされる。

なお、ヘッダー部は 1 秒ごと挿入されるが、挿入によるサンプリングデータの欠損は発生しない。

サンプリングデータ部のデータ長は [サンプリング周波数 (Hz)] × [A/D ビット数] × [入力 ch 数] (ビット) であるが、VSSP (PCI バス転送) では実用上は 64M ビット/秒が最大値であり、VSSP32 (USB2.0 転送) では、実用上の最大値は 256M ビット/秒、VSSP64 (USB3.0 転送) では実用上の最大値は 1024M ビット/秒である。

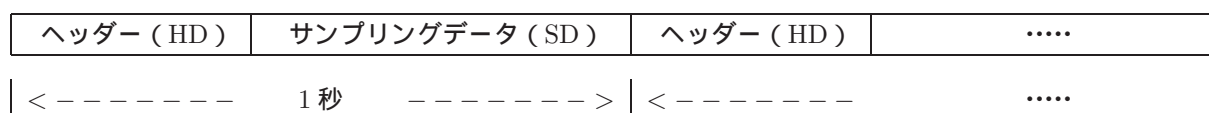


図 D.2: VSSP, VSSP32 および VSSP64 サンプラーデータ構造。

### D.2.2 VSSP ヘッダーフォーマット仕様

図 D.3 に VSSP ヘッダーフォーマット仕様を示す。ヘッダー部は 8 バイトデータから構成される。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	同期パターン (0xFFFFFFFF)																															
W1	第2同期パターン(0x8B)								AD		SFREQ				CH	時刻(0h UTCからの経過秒)(17ビット)																

AD : AD変換時のビット数インデックス 0: 1ビット / 1: 2ビット / 2: 4ビット / 3: 8ビット

SFREQ : サンプリング周波数インデックス (注: 32MHzサンプリング以上の定義は他フォーマットデータからの変換サポート用)

0: 40kHz / 1: 100kHz / 2: 200kHz / 3: 500kHz / 4: 1MHz / 5: 2MHz / 6: 4MHz / 7: 8MHz / 8: 16MHz

9: 32MHz / 10: 64MHz / 11: 128MHz / 12: 256MHz / 13: 512MHz / 14: 1024MHz / 15: 2048MHz

CH : チャンネル数インデックス 0: 1ch 1: 4ch

図 D.3: VSSP ヘッダーフォーマット (8 バイト)

<sup>3</sup>VSSP64 サンプラーは内部の DIP スイッチにより VSSP32 モードおよび VSSP64 モードを切り替える。なおオペレーション時はそれぞれのモードに応じたドライバーを使用する必要がある。

ここで

同期パターン	–	0xFFFFFFFF
第2同期パターン	–	0x8B (VSSP32 は 0x8C)
AD	–	AD 変換時のビット数インデックス (2ビット) 0: 1ビット / 1: 2ビット / 2: 4ビット / 3: 8ビット
SFREQ	–	サンプリング周波数を示すインデックス (4ビット) (注: 32MHz サンプリング以上の定義は他フォーマットデータからの変換サポート用) 0: 40kHz / 1: 100kHz / 2: 200kHz / 3: 500kHz 4: 1MHz / 5: 2MHz / 6: 4MHz / 7: 8MHz / 8: 16MHz 9: 32MHz / 10: 64MHz / 11: 128MHz / 12: 256MHz / 13: 512MHz 14: 1024MHz / 15: 2048MHz
CH	–	使用チャンネル数 0: 1ch 1: 4ch
時刻	–	00h00m00s からの経過秒数で 0 ~ 86399

### D.2.3 VSSP32 および VSSP64 ヘッダーフォーマット一般仕様

図 D.4 に VSSP32 および VSSP64 データヘッダー部の一般仕様を示す。現状の K5 ユーティリティソフトでは AUX FIELD のサイズはデフォルト値である 20 バイトを採用している。したがって、ヘッダー部全体のサイズは 32 バイトとなっている。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	同期パターン (0xFFFFFFFF)																															
W1	第2同期パターン(0x8C) (0x8D: VSSP64モード)								AD		SFREQ		CH	時刻(0h UTCからの経過秒)(17ビット)																		
W2	主バージョン#				副バージョン#				AUX FIELD サイズ (バイト) (8ビット) (デフォルト値=20)								EF ch	年 (下2桁) (6ビット)						通日 (1月1日を1日する) (9ビット)								
W3	AUX FIELD (ユーザーが自由に定義)																								AUX FIELD フォーマット#							
W4																																
W5																																
W6																																
W7																																

AD : AD変換時のビット数インデックス 0: 1ビット / 1: 2ビット / 2: 4ビット / 3: 8ビット

SFREQ : サンプリング周波数インデックス (注: VSSP32の最大値は64MHz, VSSP64の最大値は128MHz

256MHzサンプリング以上の定義は他フォーマットデータからの変換サポート用)

0: 40kHz / 1: 100kHz / 2: 200kHz / 3: 500kHz / 4: 1MHz / 5: 2MHz / 6: 4MHz / 7: 8MHz / 8: 16MHz

9: 32MHz / 10: 64MHz / 11: 128MHz / 12: 256MHz / 13: 512MHz / 14: 1024MHz / 15: 2048MHz

CH : チャンネル数インデックス 0: 1ch 1: 4ch

EF : エラーフラグ (直前のフレームでエラーが発生した時に1がセットされる)

ch : VSSP64モード時のチャンネル数インデックス 1: 2ch 0: CHフラグによる

図 D.4: VSSP32 および VSSP64 ヘッダーフォーマット一般仕様

ここで

同期パターン	– 0xFFFFFFFF
第 2 同期パターン	– 0x8C ( VSSP は 0x8B、VSSP64 モードでは 0x8D )
AD	– AD 変換時のビット数インデックス ( 2 ビット ) 0: 1 ビット / 1: 2 ビット / 2: 4 ビット / 3: 8 ビット
SFREQ	– サンプリング周波数を示すインデックス ( 4 ビット ) ( 注 : 128MHz サンプリングは VSSP64 サンプラーで可、 256M 以上の定義は他フォーマットデータからの変換サポート用 ) 0: 40kHz / 1: 100kHz / 2: 200kHz / 3: 500kHz 4: 1MHz / 5: 2MHz / 6: 4MHz / 7: 8MHz / 8: 16MHz 9: 32MHz / 10: 64MHz / 11: 128MHz / 12: 256MHz / 13: 512MHz 14: 1024MHz / 15: 2048MHz
ch	– 使用チャンネル数 0: 1ch 1: 4ch VSSP64 モード時は EF に 1 がセットされると 2ch を表わす
時刻	– 00h00m00s からの経過秒数で 0 ~ 86399
主バージョン#	– サンプラー制御 ROM バージョン主番号
副バージョン#	– サンプラー制御 ROM バージョン副番号
AUX FIELD サイズ	– AUX FIELD のサイズ ( バイト単位 ) デフォルト値は 20
EF	– エラーフラグ ( 直前のフレームでエラーが発生した時に 1 がセットされる )。 VSSP64 ではここが 1 にセットされると 2ch モード
年	– 西暦年の下 2 桁
通日	– 年内での通し日 ( 1 月 1 日を 1 日する )
AUX FIELD フォーマット番号	– AUX FIELD 部のフォーマット番号

AUX FIELD 部はユーザーが自由に定義して使用することができる。ただし、AUX FIELD 部の最初の 1 バイトをフォーマット番号と定義する。ユーザーが AUX FIELD を新たに定義する場合は、すでに定義済みまたは予約済みの番号を使用しないことが望ましい。

#### D.2.4 VSSP32 ヘッダーフォーマット番号別仕様

AUX FIELD 部の定義済みフォーマット番号および予約済み番号を表 D.4 に示す。

表 D.4: ヘッダーフォーマット定義済みおよび予約済み番号

フォーマット番号	用途
0	テスト用
1	実観測用 ( “autoobs” 出力ファイルフォーマット )
2	実観測用 ( “sampling” 出力ファイルフォーマット )
30~39	予約済み ( 宇宙研グループ )
85	テスト用
170	テスト用
21	拡張フォーマット その 1 ( 任意のサンプリング周波数および 16ch までサポート )
22	拡張フォーマット その 2 ( 任意のサンプリング周波数、AD ビット数、チャンネル数をサポート )

すでに定義済みの番号は#0, #1, #2, #85, #170, そして拡張フォーマット用の#21 および#22 である。また 30 番台は ISAS (宇宙研) グループ用に予約済みである。フォーマット#21 および#22 は任意のサンプリング周波数およびチャンネル数 (#22 では任意の AD ビット数も) をサポートするために用意されたフォーマットで、VSSP32 サンプラーからの直接出力データではなく、他フォーマットからの変換や、複数サンプラー出力の合成用途を想定したフォーマットである。

### VSSP32 フォーマット#0

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	同期パターン (0xFFFFFFFF)																															
W1	第2同期パターン(0x8C) (0x8D: VSSP64モード)								AD		SFREQ		CH	時刻(0h UTCからの経過秒)(17ビット)																		
W2	主バージョン#				副バージョン#				AUX FIELD サイズ (バイト) (8ビット) (20)								EF ch	年 (下2桁) (6ビット)						通日(1月1日を1日する) (9ビット)								
W3	すべて0																								AUX FIELD フォーマット# (0)							
W4																																
W5																																
W6																																
W7																																

図 D.5: VSSP32 ヘッダーフォーマット#0

### VSSP32 フォーマット#1

ユーティリティプログラム “autoobs” が出力するフォーマット。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	同期パターン (0xFFFFFFFF)																															
W1	第2同期パターン(0x8C) (0x8D: VSSP64モード)								AD		SFREQ		CH	時刻(0h UTCからの経過秒)(17ビット)																		
W2	主バージョン#				副バージョン#				AUX FIELD サイズ (バイト) (8ビット) (20)						EF <sub>ch</sub>	年 (下2桁) (6ビット)						通日 (1月1日を1日する) (9ビット)										
W3	局ID (2文字: 1文字の場合は1文字目は空白コード)														LPF 周波数 (MHz) 0はLPFスルー						AUX FIELD フォーマット# (1)											
W4	局名 (8文字: 無い場合はPCホスト名)																															
W5																																
W6	PCホスト名 (8文字)																															
W7																																

図 D.6: VSSP32 ヘッダーフォーマット#1

## VSSP32 フォーマット#2

ユーティリティプログラム “sampling” が出力するフォーマット。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
W0	同期パターン (0xFFFFFFFF)																																			
W1	第2同期パターン(0x8C) (0x8D: VSSP64モード)								AD				SFREQ				CH		時刻(0h UTCからの経過秒)(17ビット)																	
W2	主バージョン#				副バージョン#				AUX FIELD サイズ (バイト) (8ビット) (20)								EF ch		年 (下2桁) (6ビット)						通日 (1月1日を1日する) (9ビット)											
W3	埋め草データ(0x5555)																LPF 周波数 (MHz) 0はLPFスルー						AUX FIELD フォーマット# (2)													
W4	埋め草データ(0x55555555)																																			
W5	埋め草データ(0x55555555)																																			
W6	PCホスト名 (8文字)																																			
W7																																				

図 D.7: VSSP32 ヘッダーフォーマット#2

## VSSP32 フォーマット#85

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
W0	同期パターン (0xFFFFFFFF)																																	
W1	第2同期パターン(0x8C) (0x8D: VSSP64モード)								AD				SFREQ				CH		時刻(0h UTCからの経過秒)(17ビット)															
W2	主バージョン#				副バージョン#				AUX FIELD サイズ (バイト) (8ビット) (20)								EF ch		年 (下2桁) (6ビット)						通日 (1月1日を1日する) (9ビット)									
W3	埋め草データ(0x5555)																LPF 周波数 (MHz) 0はLPFスルー						AUX FIELD フォーマット# (85) (85=0x55)											
W4	埋め草データ(0x55555555)																																	
W5	埋め草データ(0x55555555)																																	
W6	埋め草データ(0x55555555)																																	
W7	埋め草データ(0x55555555)																																	

図 D.8: VSSP32 ヘッダーフォーマット#85

## VSSP32 フォーマット#170

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	同期パターン (0xFFFFFFFF)																															
W1	第2同期パターン(0x8C) (0x8D: VSSP64モード)								AD	SFREQ				CH	時刻(0h UTCからの経過秒)(17ビット)																	
W2	主バージョン#				副バージョン#				AUX FIELD サイズ (バイト) (8ビット) (20)								EF ch	年 (下2桁) (6ビット)						通日 (1月1日を1日する) (9ビット)								
W3	埋め草データ(0xAAAA)																LPF 周波数 (MHz) 0はLPFスルー						AUX FIELD フォーマット# (170) (170=0xAA)									
W4	埋め草データ(0xAAAAAAAA)																															
W5	埋め草データ(0xAAAAAAAA)																															
W6	埋め草データ(0xAAAAAAAA)																															
W7	埋め草データ(0xAAAAAAAA)																															

図 D.9: VSSP32 ヘッダーフォーマット#170

## VSSP32 フォーマット#21 (拡張フォーマット その1)

任意のサンプリング周波数と 16ch までのチャンネル数を許すフォーマット。従来のサンプリング周波数情報部とチャンネル数情報部は意味を持たず、新たに AUX FIELD 中でサンプリング周波数とチャンネル数を定義している。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	同期パターン (0xFFFFFFFF)																															
W1	第2同期パターン(0x8C) (0x8D: VSSP64モード)								AD	SFREQ *1				--	時刻(0h UTCからの経過秒)(17ビット)																	
W2	主バージョン#				副バージョン#				AUX FIELD サイズ (バイト) (8ビット) (20)								年 (下2桁) (7ビット)						通日 (1月1日を1日する) (9ビット)									
W3	サンプリング周波数 (MHz: 13ビット) *2												CH 数(2 <sup>n</sup> ) *3				LPF 周波数 (MHz) 0はLPFスルー						AUX FIELD フォーマット# (21)									
W4	任意のデータ(通常アスキー16文字)																															
W5																																
W6																																
W7																																

\*1: AUXフィールドのサンプリング周波数が0にセットされた時のみ有効

\*2: ここを0にセットした場合は従来のサンプリング周波数情報が有効

\*3: CH数は2<sup>n</sup>のnの値をセットする

図 D.10: VSSP32 ヘッダーフォーマット#21

ここで、チャンネル数は 2<sup>n</sup> の n で指定するが、n として許される値は 0,1,2,3,4 である。つまりチャンネル数としては最大 16 となる。AUX フィールドのサンプリング周波数は 13 ビット整数 (MHz) で表される。ここが 0 と

した場合は、従来のサンプリング周波数フィールドで記述される情報が有効となる。従って、拡張フォーマットで許される 1 MHz 以下のサンプリング周波数は 40kHz, 100kHz, 200kHz, 500kHz のみである。

### VSSP32 フォーマット#22 (拡張フォーマット その2)

任意のサンプリング周波数、任意の A/D ビット数、任意のチャンネル数を許すフォーマット。従来のサンプリング周波数情報部、A/D ビット数情報部、チャンネル数情報部は意味を持たず、新たに AUX FIELD 中でサンプリング周波数、A/D ビット数、チャンネル数を定義している。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	同期パターン (0xFFFFFFFF)																															
W1	第2同期パターン(0x8C) (0x8D: VSSP64モード)								未使用								時刻(0h UTCからの経過秒)(17ビット)															
W2	主バージョン#				副バージョン#				AUX FIELD サイズ (バイト) (8ビット) (20)								年 (下2桁) (7ビット)							通日(1月1日を1日する) (9ビット)								
W3	サンプリング周波数 (正: MHz, 負: kHz単位) *1																LPF 周波数 (MHz) 0はLPFスルー							AUX FIELD フォーマット# (22)								
W4	任意のデータ(通常アスキー14文字)																ADビット数							CH数								
W5																																
W6																																
W7																																

\*1: 16bit符号付き整数で記述。負の値の場合は絶対値がkHz単位のサンプリング周波数

図 D.11: VSSP32 ヘッダーフォーマット#22

### D.2.5 サンプラー出力データフォーマット

VSSP および VSSP32 サンプラーから出力されるサンプリングデータ部は 4 バイト (32 ビット) 単位で構成されており、VSSP も VSSP32 も同じフォーマットである。図 D.12 および D.13 に 1ch, 4ch の場合の 4 バイトデータのビット位置とサンプルデータの関係を示す。

bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1ch x 1bit sample #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1ch x 2bit sample #	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
1ch x 4bit sample #	7				6				5				4				3				2				1				0			
1ch x 8bit sample #	3						2						1						0													

図 D.12: サンプラー単体出力データフォーマット (1ch モード)

bit position		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
4ch x 1bit	ch #	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	0															
	sample #	7				6				5				4				3				2				1				0																			
4ch x 2bit	ch #	4	3			2			1			4	3			2			1			4	3			2			1			4	3			2			1										
	sample #	3								2								1								0																							
4ch x 4bit	ch #	4				3				2				1				4				3				2				1				4				3				2				1			
	sample #	1																0																															
4ch x 8bit	ch #	4								3								2								1																							
	sample #	0																																															

図 D.13: サンプラー単体出力データフォーマット (4ch モード)

## D.2.6 拡張データフォーマット その1

拡張データフォーマットその1（フォーマット#21）ではチャンネル数は1, 4以外に図 D.14, D.15, D.16 で示される2, 8, 16 チャンネルモードが許される。8ch モードでは8ビットサンプリングデータを表す際には、1 サンプルデータあたり8バイト（64ビット）必要となる。16ch モードでは4ビットサンプリングデータを表すのに8バイト（64ビット）、8ビットサンプリングデータを表すのに16バイト（128ビット）がそれぞれ必要となる。

	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2ch x 1bit	ch #	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
	sample #	15	14			13	12			11	10			9	8			7	6			5	4			3	2			1	0		
2ch x 2bit	ch #	2	1			2	1			2	1			2	1			2	1			2	1			2	1			2	1		
	sample #	7				6				5				4				3				2				1				0			
2ch x 4bit	ch #	2				1				2				1				2				1				2				1			
	sample #					3				2				1							1								0				
2ch x 8bit	ch #	2								1								2											1				
	sample #									1											2												

図 D.14: 拡張データフォーマット（2ch モード）

	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
8ch x 1bit	ch #	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1								
	sample #	3								2								1								0															
8ch x 2bit	ch #	8	7				6	5				4	3				2	1				8	7				6	5				4	3				2	1			
	sample #	1																0																							
8ch x 4bit	ch #	8				7				6				5				4				3				2				1											
	sample #	0																																							
8ch x 8bit	ch #	4								3								2								1															
	sample #	0																																							
	bit position	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32								
	ch #	8								7								6								5															
	sample #	0																																							

図 D.15: 拡張データフォーマット（8ch モード）

	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
16ch x 1bit	ch #	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																
	sample #	1																0																																															
16ch x 2bit	ch #	16				15				14				13				12				11				10				9				8				7				6				5				4				3				2				1			
	sample #	0																																																															
16ch x 4bit	ch #	8								7								6								5								4								3								2								1							
	sample #	0																																																															
	bit position	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32																																
	ch #	16								15								14								13								12								11								10								9							
16ch x 8bit	sample #	0																																																															
	bit position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
	ch #	4								3								2								1																																							
	sample #	0																																																															
	bit position	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32																																
	ch #	8								7								6								5																																							
	sample #	0																																																															
	bit position	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64																																
	ch #	12								11								10								9																																							
	sample #	0																																																															
	bit position	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96																																
	ch #	16								15								14								13																																							
	sample #	0																																																															

図 D.16: 拡張データフォーマット（16ch モード）

拡張データフォーマットにおいて、サンプリング部のデータ長は4バイト（32ビット）ワードの整数倍とする。サンプリング周波数とADビット数の兼ね合いで端数が生じるばあいは0データで埋める。1秒あたりのサンプ



リングデータ部のビット数 ( $B$ ) はサンプリング周波数を  $F$  (Hz)、AD ビット数を  $A$ 、ch 数を  $N$  とすると

$$B = F \cdot A \cdot N \quad [\text{ビット}]$$

で表されるが、 $B$  が 32 で割り切れない場合は、サンプリングデータ部のデータ数  $S$  (ビット) を次式のように定め、 $B + 1$  ビット目から残りの部分を 0 で埋める。

$$S = (\text{int}(B/32) + 1) \times 32 \quad [\text{ビット}]$$

ここで  $\text{int}()$  は切り捨てを行う関数である。 $B$  が 32 で割り切れる場合は  $S = B$  である。

### D.2.7 拡張データフォーマット その 2

拡張データフォーマットその 2 (フォーマット #22) では任意のサンプリング周波数、AD ビット数、チャンネル数が許される。データ部の同一時刻のサンプルデータは AD ビット数を  $A$ 、ch 数を  $N$  とすると  $A \cdot N$  ビットで表される。1 秒当たりのデータ部のビット数 ( $B$ ) はサンプリング周波数を  $F$  (Hz) とすると

$$B = F \cdot A \cdot N \quad [\text{ビット}]$$

で表されるが、 $B$  が 32 で割り切れない場合は、サンプリングデータ部のデータ数  $S$  (ビット) を次式のように定め、 $B + 1$  ビット目から残りの部分を 0 で埋める。

$$S = (\text{int}(B/32) + 1) \times 32 \quad [\text{ビット}]$$

$B$  が 32 で割り切れる場合は  $S = B$  である。

## D.3 Mark5B データフォーマット

### D.3.1 データ構造

Mark5B データはフレームデータから構成され、フレームデータはヘッダー部 16 バイトとサンプリングデータ部 10000 バイトの計 10016 バイトで構成されている。

ヘッダー (16bytes)	サンプリングデータ (10000 バイト)	ヘッダー (16bytes)	.....
< ----- フレーム ----- >   < ----- ..... >			

図 D.17: Mark5B データ構造。

### D.3.2 Mark5B ヘッダー部

図 D.18 に Mark5B ヘッダーフォーマット仕様を示す。ヘッダー部は 16 バイトデータから構成される。

### D.3.3 Mark5B データ部

Mark5B データ部は 10000 バイトデータで構成され、bit-stream という単位を使って図のように示される。ここで、bit-stream は 1 つのサンプリングデータの集まりを意味し、1, 2, 4, 8, 16, 32 bit-stream の 6 種類がある。チャンネル数を  $n$ 、AD ビット数を  $k$  とすると  $nk = \text{bit-stream}$  の関係がある。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	Sync Pattern : '0xABADDEED'																															
W1	User-specified																T	Frame # within second (starting at 0)														
W2	VLBA BCD Time code word 1 ('JJSSSSS')																															
W3	VLBA BCD Time code word 2 ('.SSSS')																CRC 16bit															

T 1: TVG data

図 D.18: Mark5B ヘッダー構造

1 bit-stream data																	Bit0															
Bit31	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

図 D.19: Mark5B データ構造。1 bit-stream データの場合。セル内の番号はサンプリング番号（サンプリングの順番）である。このデータは1チャンネル1ビットサンプリングに相当する。

Bit31						2 bit-stream data										Bit0				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					

図 D.20: Mark5B データ構造。2 bit-stream データの場合。このデータは1チャンネル2ビットサンプリングまたは2チャンネル1ビットサンプリングに相当する。

Bit31	4 bit-stream data																													Bit0
	7	6	5	4	3	2	1	0																						

図 D.21: Mark5B データ構造。4 bit-stream データの場合。このデータは4チャンネル1ビットサンプリングや2チャンネル2ビットサンプリングといったデータに相当する。

Bit31	8 bit-stream data																Bit0
3				2				1				0					

図 D.22: Mark5B データ構造。8 bit-stream データの場合。このデータは8チャンネル1ビットサンプリングや4チャンネル2ビットサンプリングといったデータに相当する。

Bit31	16 bit-stream data																Bit0
	1	0															

図 D.23: Mark5B データ構造。16 bit-stream データの場合。このデータは16チャンネル1ビットサンプリングや8チャンネル2ビットサンプリングといったデータに相当する。

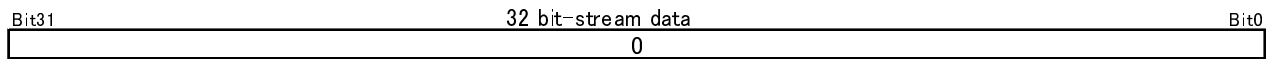


図 D.24: Mark5B データ構造。32 bit-stream データの場合。このデータは 16 チャンネル 2 ビットサンプリングといったデータに相当する。

## D.4 VDIF フォーマット

### D.4.1 データ構造

VDIF(VLBI Data Interchange Format) データはフレームデータから構成され、フレームデータはヘッダー部とサンプリングデータ部からなる(図 D.2)。ヘッダー部は 32 バイト(256 ビット)で構成され最初の 16 バイトは固定情報、残りの 16 バイトはユーザーが内容を決めることができる拡張データ部となっている。拡張データ部は現在 NICT が制定した EDV #1 と VLBA, VLA, GBT 用に制定された EDV #3 の 2 種類が制定されている。

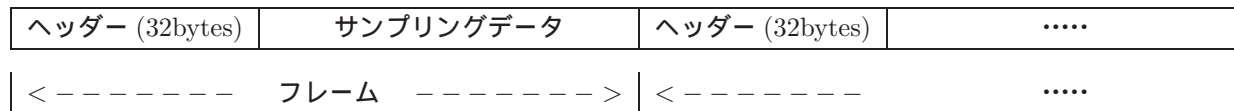


図 D.25: VDIF データ構造。

### D.4.2 VDIF ヘッダー部

#### 共通仕様

図 D.26 に VDIF ヘッダーフォーマット仕様を示す。ヘッダー部は 32 バイトデータから構成される。

	bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	I	L	Seconds from reference epoch																														
W1	Un-assign		Ref Epoch: Half year counter							Data Frame # within second																							
W2	Version Number			$\log_2(\#chns)$							Data Frame length (units of 8bytes)																						
W3	C	bits/sample-1							Thread ID for multi-stream (0-1023)										Station ID														
W4	Extended User Data Version Number									Extended User Data																							
W5	Extended User Data																																
W6	Extended User Data																																
W7	Extended User Data																																

I= Invalid Flag

L= Legacy mode 0: standard 32-byte VDIF header 1: legacy header-length (16-byte) mode

C: data type 0: real data, 1: complex data

図 D.26: VDIF ヘッダー構造

## NICT 仕様 (EDV #1)

NICT が制定した EDV #1 のヘッダーフォーマットを図 D.27 に示す。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	I	L	Seconds from reference epoch																													
W1	Un-assign		Ref Epoch: Half year counter						Data Frame # within second																							
W2	Version Number			log <sub>2</sub> (#chns)						Data Frame length (units of 8bytes)																						
W3	C	bits/sample-1						Thread ID for multi-stream (0-1023)										Station ID														
W4	Extended User Data Version Number								U	Sampling Rate (kHz or MHz)																						
W5	Sync Pattern : '0xACABFEED'																															
W6	DAS/Station Name (8 bytes)																															
W7																																

U: units of sampling rate 0: kHz, 1: MHz

図 D.27: VDIF ヘッダー構造 (NICT 仕様 EDV #1)

## VLBA, VLA, GBT 用仕様 (EDV #3)

VLBA, VLA, GBT 用ヘッダー (EDV #3) のヘッダーフォーマットを図 D.28 に示す。

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W0	I	L	Seconds from reference epoch																													
W1	Un-assign		Ref Epoch: Half year counter						Data Frame # within second																							
W2	Version Number		$\log_2(\#chns)$						Data Frame length (units of 8bytes)																							
W3	C	bits/sample-1						Thread ID for multi-stream (0-1023)										Station ID														
W4	Extended User Data Version Number										U	Sampling Rate (kHz or MHz)																				
W5	Sync Pattern : '0xACABFEED'																															
W6	LOIF Frequency Tuning Word, unsigned int - # of Hz																															
W7	Un-assigned			DBE unit						IF				Sub Band			ESB	Major Rev				Minor Rev				Personality Type						

U: units of sampling rate 0: kHz, 1: MHz

ESB: Electronic Side Band (1: upper, 0: lower)

Personality Type: x00: RFBG, x80: DDC Mark5B, x81: DDC complex, x82: DDC VDIF

図 D.28: VDIF ヘッダー構造 (VLBA, VLA, GBT 用仕様 EDV #3)

## D.4.3 VDIF データ部

## シングルチャンネルデータ例

Bit31																			real 1 bit/sample																			Bit0	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

図 D.29: VDIF データ構造。1 チャンネル 1 ビットサンプリングの場合。セル内の番号はサンプリング番号 (サンプリングの順番) である。

real 2bits/sample																Bit0
Bit31	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

図 D.30: VDIF データ構造。1 チャンネル 2 ビットサンプリングの場合。

complex 1bit/sample																																Bit0
Bit31	15Q	15I	14Q	14I	13Q	13I	12Q	12I	11Q	11I	10Q	10I	9Q	9I	8Q	8I	7Q	7I	6Q	6I	5Q	5I	4Q	4I	3Q	3I	2Q	2I	1Q	1I	0Q	0I

図 D.31: VDIF データ構造。複素 1 チャンネル 1 ビットサンプリングの場合。

complex 2bits/sample																Bit0
Bit31	7Q	7I	6Q	6I	5Q	5I	4Q	4I	3Q	3I	2Q	2I	1Q	1I	0Q	0I

図 D.32: VDIF データ構造。複素 1 チャンネル 2 ビットサンプリングの場合。

## マルチチャンネルデータ例

real 16ch X 1bit/sample																																Bit0
Bit31	CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1
<-- sample 1 -->																<-- sample 0 -->																

図 D.33: VDIF データ構造。16 チャンネル 1 ビットサンプリングの場合。

real 16ch X 2bits/sample																Bit0
Bit31	CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

図 D.34: VDIF データ構造。16 チャンネル 2 ビットサンプリングの場合。

complex 16ch X 2bits/sample															
Bit31															Bit0
CH8Q	CH8I	CH7Q	CH7I	CH6Q	CH6I	CH5Q	CH5I	CH4Q	CH4I	CH3Q	CH3I	CH2Q	CH2I	CH1Q	CH1I
CH16Q	CH16I	CH15Q	CH15I	CH14Q	CH14I	CH13Q	CH13I	CH12Q	CH12I	CH11Q	CH11I	CH10Q	CH10I	CH9Q	CH9I

図 D.35: VDIF データ構造。複素 16 チャンネル 2 ビットサンプリングの場合。

## E PGPLOT のインストール法

### E.1 FreeBSD の port を使用する方法

ネットワーク環境 ( caltech のサーバにアクセスできる ) をととのえてから、

```
cd /usr/ports/graphics/pgplot
make install
```

### E.2 ステップバイステップ法

1. /usr/local/src に pgplot5.2.tar.gz をコピー

入手先は <ftp://ftp.astro.caltech.edu/pub/pgplot/pgplot5.2.tar.gz>

2. 解凍する

```
cd /usr/local/src
gunzip -c pgplot5.2.tar.gz | tar xvof -
または tar xvzf pgplot5.2.tar.gz
```

解凍後、/usr/local/src/pgplot および サブディレクトリが出来る

3. インストール先ディレクトリ作成

```
mkdir /usr/local/pgplot
```

4. デバイスドライバーの選択

```
cd /usr/local/pgplot
cp /usr/local/src/pgplot/drivers.list .
エディター (vi 等) で選択するドライバーの先頭の"! "をはずす
vi drivers.list
```

ここで、GIF 関係のドライバーや /FILE を選択すると、FreeBSD ではエラーが出ます (linux では未確認)

おすすめドライバーの選択は

```
/LATEX /NULL /PS /VPS /CPS /VCPS /TEK4010 /RETRO /GF /GTERM /XTERM /TK4100
/VT125 /XWINDOW /XSERVE
```

5. makefile の作成

```
cd /usr/local/pgplot
FreeBSD の場合
/usr/local/src/pgplot/makemake /usr/local/src/pgplot freebsd
linux の場合
/usr/local/src/pgplot/makemake /usr/local/src/pgplot linux g77_gcc
```

6. makefile の編集

linux の場合は以下の作業 ( makefile の編集 ) は不要です

FreeBSD の場合 5 . で作った makefile で make を行くと、demo2 プログラムのコンパイル時にエラーが発生して、停止してしまう。そのため、X 端末に表示をする際に必要な pgxwin\_server が作られません。

そこで、vi 等で demo2 以降のデモプログラムをコンパイルしないように編集します

```
vi makefile
# Routine lists の中の
DEMOS= pgdemo1 pgdemo2 .....
の行の pgdemo2 以降を削除し、
DEMOS= pgdemo1
とする
```

7. make の実行

make

更に C で使用するため環境を整えるため

make cpq を実行

make clean を実行

この時点で以下のファイルがディレクトリに存在

cpqdemo grexec.f libcpqplot.a pqdemo1 pqxwin\_server cpqplot.h grfont.dat libpgplot.a pqplot.doc rgb.txt  
drivers.list grpkg1.inc makefile pqplot.inc

linux 版では更に libpgplot.so も出来ています

#### 8. ライブラリおよびCインクルードファイルを標準ディレクトリへコピー

cp libcpqplot.a /usr/lib

cp libpgplot.a /usr/lib

cp cpqplot.h /usr/include

linux では更に

cp libpgplot.so /usr/lib

#### 9. デモプログラムの実行

環境変数を以下のように設定

csh の場合 setenv PGPLOT\_DIR /usr/local/pqplot/

sh の場合 PGPLOT\_DIR="/usr/local/pqplot/"; export PGPLOT\_DIR

または export PGPLOT\_DIR=/usr/local/pqplot/

デバイスに/XWINDOW または/XSERVE を使用する場合は更に以下 (表示先) を設定

csh の場合 setenv DISPLAY IP アドレス又はマシン名:0.0

sh の場合 export DISPLAY=IP アドレス又はマシン名:0.0

pqdemo1 (FORTRAN デモ)

または

cpqdemo (C プログラムデモ) を実行する

注意事項: Tera Term から実行するときは

「Setup」→「Terminal」で「Auto switch」にチェックが入っていることを確認すること

#### 10. C プログラムのコンパイル方法以下のいずれかの方法でコンパイルする。プログラム名は test.c とする

##### 方法 1

f77 -o test test.c -lcpqplot -lpqplot -L/usr/X11R6/lib -lX11 -lm

(注: PGPLOT は元々FORTRAN で開発されたプログラムなので FORTRAN コンパイラおよびリンカーでコンパイルしないと駄目なよう。実際、cc -o test .... でコンパイル、リンクをやると undefined reference to 'xxxxxx' というエラーがたくさん出る)

##### 方法 2

cc -O2 -c -I. test.c

f77 -o test test.o -lcpqplot -lpqplot -L/usr/X11R6/lib -lX11 -lm

## F FFTW のインストール法

1. /usr/local/src に fftw-3.0.1.tar.gz をコピー

入手先は <http://www.fftw.org/fftw-3.0.1.tar.gz>

最新版は <http://www.fftw.org/> で確認のこと  
2016/10/04 現在 version 3.3.5 となっている

2. 解凍する

```
cd /usr/local/src
```

```
tar xvfz fftw-3.0.1.tar.gz
```

解凍後、/usr/local/src/fftw-3.0.1 およびサブディレクトリが出来る

3. makefile の作成

```
cd /usr/local/src/fftw-3.0.1
```

```
./configure
```

4. make の実行

```
make
```

正常に実行されれば

```
/usr/local/src/fftw-3.0.1/.libs
```

というディレクトリができる

5. ライブラリおよびC インクルードファイルを標準ディレクトリへ  
セットアップ

```
make install を実行
```

これにより、

```
/usr/local/lib 以下に
```

```
libfftw3.a および libfftw3.la
```

```
/usr/local/include 以下に
```

```
fftw3.h および fftw3.h
```

が作成される

6. C プログラムのコンパイル方法

リンク時に `-lfftw3 -lm`

を指定する。

また、`-I/usr/local/include`

`-L/usr/local/lib`

でインクルードファイルとライブラリのサーチするディレクトリを  
指定する



詳細は <http://www.fftw.org/> を参照のこと