

K5/VSI システムの手引き

木村守孝

2009年10月28日 改訂版 1.1

目次

1	K5/VSIでVSI信号を取扱うためのPC構成	2
2	CentOS 5.2のインストール	3
2.1	/etc/grub.confの編集	10
2.2	VSI2000DIMボード用デバイスドライバのインストール	11
2.3	各種アプリケーションのインストール	12
2.4	High Point製のRocket Raid 2340の認識とXFSファイルシステムのフォーマット	13
3	ソフトウェア関連器によるVERAバックアップ関連処理	14
3.1	処理の流れ	14
3.2	関連スケジュールファイルの生成	15
3.3	観測ファイルの生成	16
3.4	GICO3用遅延情報ファイルの生成	17
3.5	ソフトウェア関連処理の実行	18
3.6	関連出力ファイルのバンチング処理	19
3.7	複数の関連出力ファイルを結合する	20
3.8	関連出力ファイルをCODA型式へ変換	21
4	付録	22
4.1	ソフトの操作方法	22
4.1.1	monitor2000 (旧実行名:vsi2000_monitor)	22
4.1.2	vsi2000_time	23
4.1.3	vsi2000_rec	23
4.1.4	capture2000	24
4.1.5	gico3 (旧実行名:gico_corr)	26
4.1.6	pca12000	27
4.2	関連スケジュールの作成	28
4.2.1	異なるターミナル間での関連処理スケジュールファイル	28
4.2.2	偏波観測のための関連処理スケジュールファイル	29
4.2.3	VERA-1モードとADS1000[512Msps/2bit/01ch]の混合関連処理	31
4.3	GICO3の内部動作	32
4.4	各ファイルの命名規則	34
4.5	遅延情報ファイル構造	34
4.6	GICO3の関連処理でデータの並び	35
4.7	記録・関連スケジュール [構造]	36

1 K5/VSIでVSI信号を取扱うためのPC構成

VSI信号をVSI2000DIMボードを使用して内蔵のメモリに取り込む為には、最低限このボードが装着できる空きのPCI-Xソケットがマザーボード上にひとつ必要です。さらに、VSI信号を長時間記録する必要がある場合にはRAIDやSSD等の高速な記録装置が別途必要となります。これらの条件を満たす場合でもラックマウントサーバーでは限られたスペースにPCIボードを挿入するため、ライザーカードを使用するものがありますが、PCI信号のタイミングマージンが小さくなるため、ライザーカード経由での接続はお控え下さい。下記にNICTで推称しているシステム構成とOSを記載します。

表 1: NICTで推称するK5/VSIの構成

構成要素	メーカー等	型番等
マザーボード	Rio Works	HDAMA/HDAM-Express(販売終了)
	ASUS	P5WDG2 Pro/P5E WS
	SuperMicro	X7DBE (現行 K5/VSI で使用)
メモリ		4GB
RAID カード	High Point	Rocket Raid 1820/1820A/2340(現行 K5/VSI で使用)
	Areca	ARC-1260
OS		CentOS 5.2 for x86-64

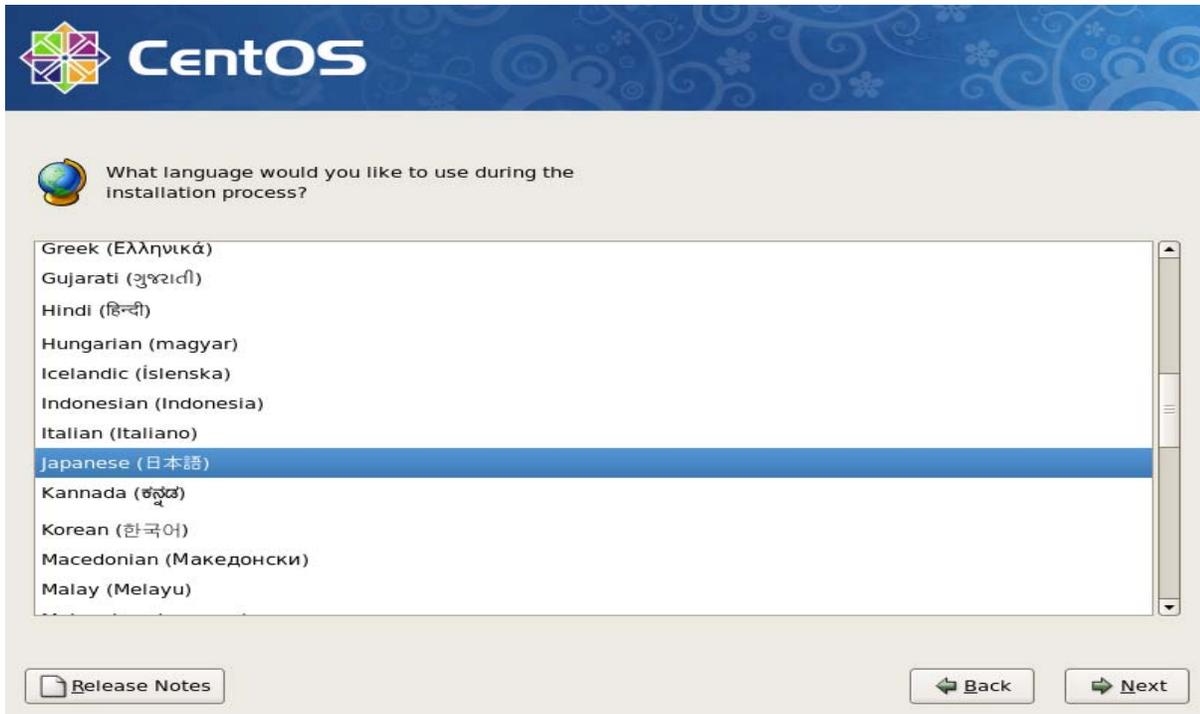
2 CentOS 5.2 のインストール

CentOS のインストール自体は通常の手順通りに行なえば問題はありません。参考までに NICT でのインストール設定等を紹介しします。

初期画面 ⇒ Next



インストール時の言語選択 Japanese(日本語)⇒ Next



キーボードの選択 日本語 ⇒ Next



インストール インストール (I) ⇒ 次



パーティション設定 カスタムレイアウト ⇒ 次



パーティション設定 ルート=16GB swap=4GB 残り=home ⇒ 次



CentOS

ドライブ /dev/hde (238473 MB) (モデル: ASI ARAID99 1010 Rev 1.2)

hde1	hde3	
16002	218469	MB

新規(W) 編集(E) 削除(D) リセット(S) RAID(A) LVM(L)

デバイス	マウントポイント/ RAID/ボリューム	タイプ	フォーマットする	容量 (MB)	開始	終了
▼ ハードドライブ						
▼ /dev/hde						
/dev/hde1	/	ext3	✓	16002	1	2040
/dev/hde2		swap	✓	4000	2041	2550
/dev/hde3	/home	ext3	✓	218469	2551	30401

RAID デバイス/LVM ボリュームグループメンバーを非表示にする(G)

リリースノート(R)
戻る(B)
次(N)

ブートローダ デフォルト ⇒ 次



CentOS

GRUB ブートローダは、/dev/hde 上にインストールされます。

ブートローダはインストールされません。

他のオペレーティングシステムがブートできるようにブートローダを設定できます。これにより、一覧からブートするオペレーティングシステムを選択できるようになります。自動的に認識されない他のオペレーティングシステムを追加するには、「追加」をクリックします。デフォルトでブートするオペレーティングシステムを変更するには、目的のオペレーティングシステムで「デフォルト」を選択します。

デフォルト	ラベル	デバイス
<input checked="" type="checkbox"/>	CentOS-5.3	/dev/hde1

追加(A)
編集(E)
削除(D)

ブートローダパスワードによってカーネルに渡されるオプションをユーザが変更してしまうのを防ぎます。より高度なシステムセキュリティを確保するために、パスワードを設定することを推奨します。

ブートローダパスワードを使用(U) パスワードを変更(P)

高度なブートローダオプションの設定(Q)

リリースノート(R)
戻る(B)
次(N)

ネットワーク 適宜設定 ⇒ 次

 **CentOS**

ネットワークデバイス

起動時にアクティブ	デバイス	IPv4/ネットマスク	IPv6/プレフィックス
<input checked="" type="checkbox"/>	eth0	DHCP	Auto
<input type="checkbox"/>	eth1	DHCP	Auto

[編集\(E\)](#)

ホスト名
ホスト名を設定:

DHCP経由で自動設定(A)

手動設定(M) (例、 host.domain.com)

その他の設定

ゲートウェイ(G):

1 番目の DNS(P):

2 番目の DNS(S):

[リリースノート\(R\)](#) [戻る\(B\)](#) [次\(N\)](#)

地域 デフォルト ⇒ 次

 **CentOS**

マップ内で地域を選んでクリックしてください:



システムクロックで UTC を使用 (S)

[リリースノート\(R\)](#) [戻る\(B\)](#) [次\(N\)](#)

ルートアカウント 適宜設定 ⇒ 次



CentOS

root アカウントはシステムの管理に使用します。 root ユーザーのパスワードを入力してください。

Root パスワード(P):

確認(C):

[リリースノート\(R\)](#) [戻る\(B\)](#) [次\(N\)](#)

追加タスク 今すぐカスタマイズ ⇒ 次



CentOS

CentOS のデフォルトインストールには全般的なインターネット使用に適用できるソフトウェアのセットが含まれています。システムにサポートさせる追加タスクを指定してください。

- Desktop - Gnome
- Desktop - KDE
- Server
- Server - GUI

ソフトウェアのインストールに使用する予定の追加リポジトリを選択してください。

- Packages from CentOS Extras

[追加でソフトウェアリポジトリを加える \(A\)](#)

ここで、ソフトウェア選択をさらに詳細にカスタマイズすることができます。また、インストール完了後にソフトウェア管理アプリケーションから行うことも可能です。

後でカスタマイズする(L) **今すぐカスタマイズする(C)**

[リリースノート\(R\)](#) [戻る\(B\)](#) [次\(N\)](#)

仮想化/クラスタリング/クラスタストレージ以外は全て選択 ⇒ 次



インストール開始 ⇒ 次



インストール完了後、ユーザーの登録とセキュリティポリシーの設定等を行えば OS のインストールは完了です。

2.1 /etc/grub.confの編集

VSI2000DIM ボードを使って VSI データを PC へ転送するためには、PC 上の物理メモリ空間で連続した領域を OS が起動する以前に確保する必要があります。Linux では起動オプション "mem=" を設定することにより、kernel が使用するメモリアドレスの上限を制限することが出来ます。例えば 4GB のメモリがある場合には、オプション "mem=2048M" を設定することでアドレス空間が 0x80000000-0xFFFFFFFF の 2GB の領域は kernel から利用されなくなり、VSI2000DIM ボードがこのメモリを内部バッファとして利用することが出来ます¹。下記に起動オプション "mem=1920M" ²を追加した Linux を 2 番目に追加し、自動的に 2 番目から起動されるように default=1³とした "/etc/grub.conf" を以下に記載します。なお、ここでの設定は OS を再起動することにより有効になります。

```
***** /etc/grub.conf ここから *****

# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You do not have a /boot partition. This means that
#           all kernel and initrd paths are relative to /, eg.
#           root (hd0,0)
#           kernel /boot/vmlinuz-version ro root=/dev/hde1
#           initrd /boot/initrd-version.img
#boot=/dev/sda
default=1      #0 から 1 に変更
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-92.el5)
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.18-92.el5 ro root=LABEL=/ rhgb quiet
    initrd /boot/initrd-2.6.18-92.el5.img
#2 番目の起動カーネルに mem=1920M を追加
title CentOS (K5VSI)
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.18-92.el5 ro root=LABEL=/ rhgb quiet mem=1920M
    initrd /boot/initrd-2.6.18-92.el5.img

***** /etc/grub.conf ここまで *****
```

¹kernel が高位のアドレスに PCI のメモリ空間をマッピングする事があるので、実際には 0xFFFFFFFF までは使用できません

²緩衝領域として 2048M より少なめの値を設定しています

³ここではゼロから数えます

2.2 VSI2000DIM ボード用デバイスドライバのインストール

K5/VSI用に配付されている VSI2000DIM 用のデバイスドライバを以下のように rpm コマンドを使いインストールします⁴。

```
rpm -ivh --force k5vsi_driver-2.4.x-x.x86_64.rpm
```

正常にドライバがインストールされた後に、mknod コマンドで以下のようにしてデバイスファイル”/dev/k5vsi”を作成します

```
/sbin/mknod -m 666 /dev/k5vsi c 250 0
```

この後ドライバをカーネル内にロードしますが、ここで 1 秒分のバッファ長やバッファ数、バッファオフセットアドレス等を指定することができます。前述の/etc/grub.conf で”mem=1920M”を指定し、1024Mbit(約 0x08000000Btye)のバッファを 8 枚分使用する時には以下のような設定をします。

```
/sbin/insmod /lib/modules/2.6.18-92.el5/kernel/drivers/misc/k5vsi.ko Offset=0x80000000 Length=0x08000000 Number=8
```

ドライバが正常にカーネルにロードされた状態で 32MHz の VSI 信号を入力すると、VSI2000DIM ボード上の LED が規則的に点滅し、ボードが正常に動作していることを確認できます。この方法では毎回起動後に上記の設定をしなくてはならないため、正常にドライバの動作が確認出来た場合には以下のように/etc/rc.local に追加します。

```
***** /etc/rc.local ここから *****

#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

/bin/mknod -m 666 /dev/k5vsi c 250 0
/sbin/insmod /lib/modules/2.6.18-92.el5/kernel/drivers/misc/k5vsi.ko Offset=0x80000000 Length=0x08000000 Number=8

***** /etc/rc.local ここまで *****
```

上記の設定ではバッファサイズが 0x08000000 バイトであるため、1024Mbps の VSI 信号のみを取扱うことが出来ます。もし 2048Mbps の VSI 信号も取扱う場合には Length の値を 0x10000000 として、Number の値を半分の 4 とします。ここでの指定では 0xBFFFFFFF までのメモリしか使用していませんが、より高位のメモリを使用する場合には、他の PCI デバイスが使用する空間と競合が起きないように/sbin/lspci -vvv などの情報を確認して注意深く設定する必要があります。

VSI2000DIM ボードを内蔵していない PC で試験的にソフトウェアの動作実験等を行う場合には、デバイスのロード時に”Mode=Emulation”を追加することで擬似的に VSI2000DIM ボードが存在するかのように振る舞うことが可能です。この場合には VSI 信号のクロック周波数情報はどこにも存在しないので、オプションで VSI 信号のクロック周波数を指定する必要があります。Clock を指定しない場合は 32MHz(1024Mbps)での動作となりますが、”Clock=64000000”とすれば 64MHz(2048Mbps)での動作となります。この場合のクロック周波数は Length の値により最大値は制限を受けますが、最小値は任意の値が設定できます。

⁴最新版は K5/VSI のホームページ (<http://www2.nict.go.jp/w/w114/stsi/K5/VSI/K5VSI/index.html>) からダウンロードできます

2.3 各種アプリケーションのインストール

VSI2000-DIM用に配付されている最新のアプリケーションソフトをRPMコマンドを使いインストールします。ソフトウェアのバージョンアップや新規で追加されることがありますので、最新版はホームページ (<http://www2.nict.go.jp/w/w114/stsi/K5/VSI/K5VSI/index.html>)でお確かめ下さい。

```
rpm -ivh --force vsi2000_tool-2.4.5-1.x86_64.rpm #記録用基本ソフト
rpm -ivh --force monitor2000-2.4.1-1.x86_64.rpm #信号モニターソフト
rpm -ivh --force capture2000-2.4.5-1.x86_64.rpm #スケジュール記録ソフト
rpm -ivh --force pcal2000-2.4.0-1.x86_64.rpm #PCAL 検出&バンドパス表示
rpm -ivh --force skd2xml-1.0.0-1.x86_64.rpm #スケジュール変換ソフト SKD->XML
rpm -ivh --force gico3-2.4.5-1.x86_64.rpm #関連処理ソフト
rpm -ivh --force fringe-2.4.4-1.x86_64.rpm #フリンジサーチソフト
rpm -ivh --force cor2cout-1.0.0-1.x86_64.rpm #関連出力ファイル変換ソフト
rpm -ivh --force mtkfx-1.0.3-1.x86_64.rpm #VERA 用変換ソフト
rpm -ivh --force fftwf-3.2.1-1.x86_64.rpm #FFT ライブラリ
rpm -ivh --force gnuplot-4.0.0-1.x86_64.rpm #表示ソフト
```

2.4 High Point 製の Rocket Raid 2340 の認識と XFS ファイルシステムのフォーマット

K5/VSIでの VSI 信号のデータ記録は大容量ファイルの記録に適したファイルシステムを使用する必要があります。また、予期せぬディスク記録速度の低下によりファイル中のデータ時系列の乱れを回避するために、K5/VSIのデータ記録には Sparse-File に対応していることを前提とします。現在 Spares-File に対応したファイルシステムには XFS や Ext4 等がありますが、K5/VSI では XFS を使用することを推奨しています。RAID カードはユーザーが任意のものを選択することが出来ますが、ここでは推奨の High-Point 製 RR2340 を使用する時の設定を紹介します。PC に電源を投入した直後、RAID の BIOS 画面が表示されている時に CTRL-H を押し、表示画面にしたがって RAID-0 型のアレイを作製します⁵。推称環境である CentOS-5.2(64bit) を使用している場合には、OS の起動後に K5/VSI 用に配付しているパッケージを以下の rpm コマンドでインストールすることによりドライバソフトのコンパイル作業を省略することが出来ます。

```
rpm -ihv rr2340-x.x.x-x.x86_64.rpm
```

最新のドライバを使用したい場合には、製造メーカのホームページ”<http://www.highpoint-tech.com/>” からソースコードをダウンロードして、ソースコード内の Readme 等を参考にインストールして下さい。ドライバのインストール後に OS を再起動すると、RAID が正常に認識されている場合には dmesg コマンドで以下のようなメッセージを確認することが出来ます。以下のメッセージでは 1TB のディスク 16 台から構成される約 16TB の RAID が sdb として検出されています。

```
rr2340:RocketRAID 2340 controller driver v1.5.08.0710 (Apr 7 2009 16:08:11)
SCSI device sdb: 31253856256 512-byte hdwr sectors (16001974 MB)
```

K5/VSI で推奨している XFS は CentOS の標準パッケージ構成ではインストールされないので、K5/VSI 用に配付しているパッケージを以下の rpm コマンドでインストールします。

```
rpm -ihv kmod-xfs-0.4-1.2.6.18_92.el5.x86_64.rpm
rpm -ihv xfsprogs-2.9.4-1.el5.centos.x86_64.rpm
rpm -ihv xfsprogs-devel-2.9.4-1.el5.centos.x86_64.rpm
```

パッケージが正常にインストールされた後に下記の一連の操作で RAID のフォーマットとマウントを行ないます。

```
mkfs.xfs /dev/sdb #sdb として RAID が見えていた場合
mkdir /mnt/raid
mount -t xfs /dev/sdb /mnt/raid
```

このままでは起動時に毎回 mount コマンドを実行する必要がため、正常に RAID への読み書きが実行出来ることを確認した段階で、/etc/fstab に以下の一行を追加することにより自動でマウントされるようになります。

```
/dev/sdb          /mnt/raid        xfs      defaults    0 0
```

⁵詳細は High Point 社の RR2340 用ユーザーマニュアルをご覧ください

3 ソフトウェア相関器による VERA バックアップ相関処理

3.1 処理の流れ

ソフトウェア相関器で VERA 観測の相関処理を行う場合には、一旦磁気テープからハードディスク上に観測ファイルとして観測データのコピーを行います。相関器はこれらの観測ファイルと遅延情報ファイルと相関スケジュールファイルを使用して一連の相関処理を実行します。相関処理に必要な遅延情報ファイルと相関スケジュールファイルは三鷹 FX で使用される遅延情報ファイルと観測局で生成される観測ログファイルから変換ソフトにより生成します。相関処理後に得られる相関出力ファイルは GICO3 形式のため、幾つかの変換ソフトを使用して、最終的に VERA 用の解析系で使用される CODA ファイル形式に変換されます。以下に一連の処理の流れと各処理で使用されるコマンド名を列挙します。

- 観測ログから相関スケジュールファイルを生成する (コマンド名 : log2xml)
- 相関スケジュールファイルを利用して観測テープから観測ファイルにコピーを行なう (コマンド名: capture2000)
- 三鷹 FX 用遅延情報ファイルから GICO3 用遅延情報ファイルを生成する (コマンド名 : tau2geo)
- 相関スケジュールファイルを利用してソフトウェア相関処理を行なう (コマンド名 : gico3)
- 個々の相関処理結果ファイルに対してバンチング処理を行なう (コマンド名 : bunch)
- 個々のバンチング済相関出力ファイルに対して基線単位で全スキャンを結合する (コマンド名 : join)
- 基線毎の相関出力ファイルを CODA 型式に変換する (コマンド名 : cor2dat)

3.2 関連スケジュールファイルの生成

観測ログファイルから関連スケジュールファイルを生成する為には変換ソフト `log2xml` を使用します。このソフトは下記のように複数の観測ログを指定することにより、多基線用の関連スケジュールを標準出力に表示しますので、出力先をファイルにリダイレクトすることによりスケジュールファイルをできます。

例 : `log2xml MIZNAO20.log IRIKI.log OGASA20.log ISHIGAKI.log > schedule.xml`

VERA バックアップ関連処理では関連出力速度を三鷹 FX の 0.8 秒と厳密に一致させるため、オプションに `-pp=5` を追加して関連出力速度を 5Hz に設定します。関連終了後に関連結果ファイルを 4PP ずつ再積分をすることにより擬似的に 0.8 秒積分相当の処理結果を得ます。デフォルトの FFT 数は全ストリーム共通で 2048 点となっていますので、これを変更する場合にはオプションに `-fft=points` を追加します。さらに、ストリーム毎に FFT 数を変えて関連処理する必要がある場合には、生成されたスケジュールファイルを直接編集することで対応が可能です。VERA の観測ログ中に西暦情報は存在しないため、この変換ソフトでは実験コード名の 2 文字目と三文字目を西暦の下 2 桁と仮定し、それに 2000 を足したものを西暦として取り扱います。実験コード名が想定されたフォーマットでない等の場合にはオプション `-year=YYYY` で直接西暦情報を指定することが出来ます。

```
##### コマンドヘルプ #####
名前
    log2xml - VERA 観測ログから GIC03 用スケジュールを生成します
書式
    log2xml VERA 観測ログ VERA 観測ログ... --fft=int --pp=int --year=int
    [--help] [--version]
オプション
    --fft          FFT 数を指定します
    --pp           関連出力速度 [Hz] を指定します
    --year         西暦を直接指定します
    --help         この画面を表示します
    --version      バージョンとコンパイルされた日時を表示します
#####
```

3.3 観測ファイルの生成

観測テープから観測ファイルを生成するためには capture2000 と VERA のテープ再生ソフトを同時に使用します。capture2000 のファイル生成の順番はスケジュールファイルにより決まっていますので、VERA のテープ再生ソフトはスケジュールファイル内に記述のある schedule⇒capture の順番でテープを再生する必要があります。テープの再生時刻は TCP 通信により VERA の制御系から入手する必要があるため、使用する IP アドレスとポート番号を起動オプション-VERA で指定します。観測ファイルを記録するディレクトリは-raw-file オプションで指定することでき、通常”/mnt/raid” と指定します。ソフトの起動後、メニューからファイル → オープンを選択して関連スケジュールファイルを読み込むと、GUI 上のスケジュール画面にスケジュールが表示されます。VERA 再生系と K5/VSI システムを同期させるために、GUI 画面上の同期制御の中から VERA 制御系を選択します⁶。また、テープコピーでは再生時刻が時系列順とはならない場合があるため、記録モードにはコピーモードを選択します。記録モード中の”自動時刻”と”非同期”はデバッグ用ですのでかならず非選択状態とします。最後に VERA のテープ再生ソフトをスタートさせた後に、capture2000 の記録開始ボタンを押してファイルにコピーを開始します。スケジュールの途中からコピーを再開する場合には、GUI 上の記録スケジュールの左側のチェックリストを非選択にすればそのコピーは無視され途中からテープコピーが続行されます。

```
##### コマンドヘルプ #####
名前
    capture2000 - スケジュールに従ってファイルへ記録します
書式
    capture2000 --schedule=string --raw-file=string
                --VERA=string [--help] [--version]
オプション
    --schedule      スケジュールファイルを指定します
    --raw-file      記録先のディレクトリ名を指定します
    --VERA          テープ時刻に同期させます [string=address:port]
    --debug         デバッグ情報を表示します
    --help          この画面を表示します
    --version       バージョンとコンパイルされた日時を表示します
#####
```

⁶K5/VSI と再生系の時計が異なり、再生系の時計が 3 秒間に時系が安定している場合に K5/VSI の再設定が行われます

3.4 GICO3 用遅延情報ファイルの生成

GICO3 の相関処理では遅延情報を内部で計算することも出来ますが、VERA 用の相関処理では三鷹 FX 用の遅延情報ファイルから tau2geo を使用して変換される GICO3 用の遅延情報ファイルを使用します。この変換は各観測局毎に全てのチャンネルについて行ないます。この変換ソフトで指定するラベル名は log2xml で生成されるスケジュールファイルに合わせ”CH01”のようにチャンネル番号をラベル名とします。例として、三鷹 FX 用の遅延情報ファイル”MIZNAO20.A.taug”、”MIZNAO20.B.taug” から GICO3 用の遅延情報ファイル”MIZNAO20_YYYYDDDHHMMSS_CH01.geo”、”MIZNAO20_YYYYDDDHHMMSS_CH02.geo” を作成する場合のコマンドを下記に記載します。

```
tau2geo -input=MIZNAO20.A.taug -station=MIZNAO20 -label=CH01
tau2geo -input=MIZNAO20.B.taug -station=MIZNAO20 -label=CH02
```

コマンド実行後、MIZNAO20_YYYYDDDHHMMSS_CH01.geo 等がカレントディレクトリに生成されますので、gico3 の実行時に指定する遅延情報ファイル用のディレクトリにファイルを移動させます。⁷

```
##### コマンドヘルプ #####
名前
    tau2geo - 三鷹 FX 形式の遅延情報ファイルを GICO3 形式へ変換します
書式
    tau2geo --input=string --station=string --label=string
        <--help> <--version>
オプション
    --input      三鷹 FX 形式の遅延情報ファイルを指定します
    --station    遅延量の計算に使用した観測局名を指定します
    --label      遅延量の計算に使用した天体名又はラベル名を指定します
    --verbose    詳細な情報を表示します
    --help      この画面を表示します
    --version    このソフトのバージョンとコンパイルされた日時を表示します
#####
```

⁷-station=./geo-file/MIZNAO20 等とするとディレクトリ名 geo-file に直接遅延情報ファイルが生成されます

3.5 ソフトウェア相関処理の実行

ソフトウェア相関器 gico3 では log2xml で生成した相関スケジュールファイルと、capture2000 で取得した全局の観測データファイルと tau2geo で変換した全局/全チャンネルの遅延情報ファイルを使用して相関処理を実行します。カレントディレクトリ上に相関スケジュールファイル”./schedule.xml” と GICO3 用の遅延情報ファイルを格納したディレクトリ”./geo-file/” と相関結果が保存されるディレクトリ”./cor-file” を用意した後、以下のコマンドで相関処理を実行します。デフォルトの設定ではフリッジ回転補正を FFT の後に行なっていますが、-complex オプションを指定することにより、FFT の前段でフリッジ回転処理を行なうことが出来ます。相関処理の具体的な内部動作は付録を参照して下さい。フリッジ検出の確認のために一部のデータの相関処理をのみ実行する場合には、-epoch と -length で相関処理開始時刻と、相関処理時間を設定します。相関処理が完了すると”./cor-file” 内に各基線毎のファイルが生成されていますので、fringe を使用して 2 次元フリッジサーチを行い、フリッジが検出されているかを確認する事が出来ます。

```
gico3_corr --schedule=schedule.xml --raw-file=/mnt/raid --geo-file=./geo-file --cor-file=./cor-file
```

```
##### コマンドヘルプ #####
```

名前

```
gico3_corr - スケジュールファイルに従い相関処理を実行します
```

書式

```
gico3_corr --schedule=string --raw-file=string
--geo-file=string --cor-file=string --multi=int
--epoch='YYYY/DDD HH:MM:SS' --length=int
--complex --command=string [--help] [--version]
```

オプション

```
--schedule      GICO3 用の相関スケジュールファイルを指定します
--raw-file       観測ファイルの格納ディレクトリ名を指定します
--geo-file       遅延ファイルの格納ディレクトリ名を指定します
--cor-file       相関結果ファイルの格納ディレクトリ名を指定します
--epoch         [epoch:epoch+length-1] 間の相関処理のみを実行します
--length        [epoch:epoch+length-1] 間の相関処理のみを実行します
--multi         相関処理で使用するスレッド数を指定します
--boost         相関処理を間引きして実行速度を見掛け上向上させます
--complex       FFT 前にフリッジ回転処理をおこないます
--command       1 秒毎に指定したスクリプトファイルを起動します
--verbose       相関処理中に詳細を表示します
--help         この画面を表示します
--version       ソフトのバージョンとコンパイルされた日時を表示します
```

```
#####
```

3.6 相関出力ファイルのバンチング処理

三鷹 FX 相関器にあるバンチング相当の機能は GICO3 では相関処理後にコマンド bunch により実行されます。この bunch は時間領域での積分と周波数領域での積分を同時に行うことができます。バンチング処理のフォーマットは PP 積分数の後に',' と三鷹 FX 用のバンチング用フォーマットを追加したもので、オプション-pattern='04,00000-01023,16,16' 等と指定します。バンチング処理されたデータはオプション-output で指定されたファイルに出力されます。オプション-output を指定しない場合は、-input で指定されたファイル名のラベル部にバンチングパターンを追加したものを出力ファイル名とします。なお、GICO3 による相関処理での FFT 点数は可変なので、最終的に三鷹 FX と同じ周波数点数を得るためには、三鷹 FX とは異なるパターンを指定する必要がある場合があります。

例 1 : 水沢-水沢基線の開始時刻"2003/328 01:34:00" の CH01 を周波数領域 0-1023 までを 16 分の 1 に積分する
bunch -input=MIZNAO20_MIZNAO20_2003328013400_CH01.cor -pattern="01,00000-01023,16,16"

例 2 : カレントディレクトリにある相関出力ファイル全てを時間領域で 4 分の 1、周波数領域で 16 分の 1 に積分する
find . -name '*.cor' -exec bunch -pattern='04,00000-01023,16,16' -input {} \;

```
##### コマンドヘルプ #####
名前
    bunch - GICO3 の相関出力ファイルを時間・チャンネル積分します
書式
    bunch --input=string --output=string --pattern='int,int-int,int,int'
        [--help] [--version]
オプション
    --input      入力用の GICO3 型式の相関出力ファイル名を指定します
    --output     出力用の GICO3 型式の相関出力ファイル名を指定します
    --pattern    バンチングパターンを指定します 例'01,00000-01023,16,16'
    --help      この画面を表示します
    --version    このソフトのバージョンとコンパイルされた日時を表示します
#####
```

3.7 複数の相関出力ファイルを結合する

GICO3の相関出力ファイルは各スキャン/各基線/各チャンネル毎に出力されていますが、これと違いCODA型式の相関出力ファイルはスキャン毎に分割されてはならず、ひとつの出力ファイル中に全スキャンが時系列順になっています。そのため、GICO3の相関出力をCODA型式へ変換する場合は、一旦全スキャンをjoinコマンドを使用してひとつに結合させたGICO3型式の相関出力ファイルを作成する必要があります。この結合では指定されたファイルの順番で結合されるので、時刻が古いものほど先に指定する必要があります⁸。このソフトは最初に指定された相関出力ファイルに2番目以降の相関出力ファイルのセクター部を追加するよう設計されているため、異なるFFT数を持つ相関出力ファイルの結合は不可能であり、また2番目以降の天体情報などは失われます⁹。

例1:水沢-水沢基線の全スキャンからチャンネルCH01をMIZNAO20_MIZNAO20_YYYYDDDDHHMMSS_CH01.corに結合する

```
join -output=MIZNAO20_MIZNAO20_YYYYDDDDHHMMSS_CH01.cor MIZNAO20_MIZNAO20*_CH01.cor
```

```
##### コマンドヘルプ #####
名前
    join - 複数の GICO3 用相関出力ファイルをひとつのファイルに結合します
書式
    join  --output=string FILE-1 FILE-2 [--help] [--version] ...
引数
    FILE-1 ...      結合する GICO3 用相関出力ファイルを複数指定します
オプション
    --output        結合後の GICO3 型式の相関出力ファイルを指定します
    --help          この画面を表示します
    --version       このソフトのバージョンとコンパイルされた日時を表示します
#####
```

⁸相関出力ファイルの時刻部分を'*'等で指定した場合、自動的に時系列順に並べられます

⁹相関結果ファイルのフォーマット参照

3.8 相関出力ファイルを CODA 型式へ変換

全スキャンのデータをひとつに結合した GIC03 型式の相関処理ファイルを cor2dat コマンドを使用して CODA 型式に変換します。この変換時にオプション `--output` を指定しない場合には入力ファイルの拡張子を ".dat" にしたもののへ出力を行いません。

例 1：水沢-水沢基線の全スキャン分の CH01 を CODA 型式に変換する

```
cor2dat --in=MIZNA020_MIZNA020_YYYYDDDDHHMMSS_CH01.cor
```

例 2：カレントディレクトリ以下にある全スキャン分の相関出力ファイルを変換する

```
find . -name '*YYYYDDDDHHMMSS*.cor' -exec cor2dat --input {} \;
```

```
##### コマンドヘルプ #####
名前
    cor2dat - GIC03 型式の相関出力ファイルを CODA 型式の相関出力ファイルへ変換します
書式
    cor2dat --input=string --output=string [--help] [--version]
オプション
    --input      GIC03 型式の相関結果ファイルを指定します
    --output     CODA 型式の相関出力ファイルを指定します
    --help      この画面を表示します
    --version    このソフトのバージョンとコンパイルされた日時を表示します
#####
```

4 付録

4.1 ソフトの操作方法

4.1.1 monitor2000 (旧実行名:vsi2000_monitor)

VSI2000DIM のモニタソフトである monitor2000 は現在のボードの状態と P-DATA の内容、そしてバッファの内容をリアルタイムで表示します¹⁰。このソフトは他の記録ソフトとの同時起動ができますので、観測中は VSI 信号の確認用に立ち上げ状態にしておいて下さい。低スペックの PC を使用している場合は観測中の信号監視には TVG チェック機能による負荷をなくすためにオプション”-noTVG” を使用した方が記録が安定する場合があります。

The screenshot shows the monitor2000 interface with a table of data. The table has five columns: Physical Memory, Epoch@Speed, Physical Memory Contents, and TVG Check. The data is organized into rows, each representing a different memory buffer. Red circles with numbers 1 through 6 are placed around the table to highlight specific information: 1 points to the Epoch@Speed column, 2 points to the TVG Check column, 3 points to the Physical Memory column, 4 points to the Epoch@Speed column, 5 points to the Physical Memory Contents column, and 6 points to the TVG Check column.

Physical Memory	Epoch@Speed	Physical Memory Contents	TVG Check
0x80000000-0x87a11fff	2003/004 01:50:24@1024Mbps	0xffffffff-0x00000000-0x03000401-0x32180000 ...	0x00000000 +000.0000006 [%]
0x88000000-0x8fa11fff	2003/004 01:50:25@1024Mbps	0xffffffff-0x00000000-0x03000401-0x32190000 ...	0x00000000 +000.0000008 [%]
0x90000000-0x97a11fff	2003/004 01:50:26@1024Mbps	0xffffffff-0x00000000-0x03000401-0x321a0000 ...	0x00000000 +000.0000008 [%]
0x98000000-0x9fa11fff	2003/004 01:50:19@1024Mbps	0xffffffff-0x00000000-0x03000401-0x32130000 ...	0x00000000 +000.0000008 [%]
0xa0000000-0xa7a11fff	2003/004 01:50:20@1024Mbps	0xffffffff-0x00000000-0x03000401-0x32140000 ...	0x00000000 +000.0000006 [%]
0xa8000000-0xafa11fff	2003/004 01:50:21@1024Mbps	0xffffffff-0x00000000-0x03000401-0x32150000 ...	0x00000000 +000.0000008 [%]
0xb0000000-0xb7a11fff	2003/004 01:50:22@1024Mbps	0xffffffff-0x00000000-0x03000401-0x32160000 ...	0x00000000 +000.0000008 [%]
0xb8000000-0xbfa11fff	2003/004 01:50:23@1024Mbps	0xffffffff-0x00000000-0x03000401-0x32170000 ...	0x00000000 +000.0000010 [%]

- 1:VSI2000DIM の現在時刻と動作速度が表示されます (ここに表示される時刻がすべての時間の基本となります)
- 2:1 秒前の P-DATA の内容が表示されます (P-DATA が無い場合は何も表示されません)
- 3:各バッファが使用するメモリ空間を表示します
- 4:各バッファに記録された観測時刻とその時のデータ速度を表示します
- 5:各バッファに記録された VSI データの最初の 4 ワードと最後の 1 ワードを表示します
- 6:TVG 信号との一致度を表示します

```
##### コマンドヘルプ #####
名前
    vsi2000_monitor - K5VSI の現在状況を表示します
書式
    vsi2000_monitor --noTVG [--help] [--version]
オプション
    --noTVG      TVG チェック機能を無効にします
    --help      この画面を表示します
    --version    バージョンとコンパイルされた日時を表示します
#####
```

¹⁰このソフトは /dev/mem にアクセスするため root での起動が必要です

4.1.2 vsi2000_time

vsi2000_time コマンドを使用することで、K5/VSIのデバイス時刻を指定時刻に設定することが出来ます。K5/VSIの時刻はコマンドを発行した瞬間に設定され、その後の 1PPS 信号を検出した度に 1 秒ずつ繰り上がります。

```
##### コマンドヘルプ #####
名前
    vsi2000_time - K5VSI のデバイス時刻を設定します
書式
    vsi2000_time --set='YYYY/DDD HH:MM:SS' --ads1000
                --now --show=int <--help> <--version>
オプション
    --set          デバイス時刻を 'YYYY/DDD HH:MM:SS' に設定します
    --ads1000      デバイス時刻を ADS1000 の P-DATA に同期させます
    --now          デバイス時刻を計算機時刻に設定します
    --show         デバイス時刻を n 秒間表示します
    --help         この画面を表示します
    --version      バージョンとコンパイルされた日時を表示します
#####
```

4.1.3 vsi2000_rec

vsi2000_rec コマンドを使用することで、指定した任意の時刻から VSI 信号をディスク上に記録することが出来ます。この時ディスク上に記録されるファイル名はオプション-fileで指定した文字列に記録開始時の時刻と拡張子”_YYYYDDDDHHMMSS.raw”が追加されます。

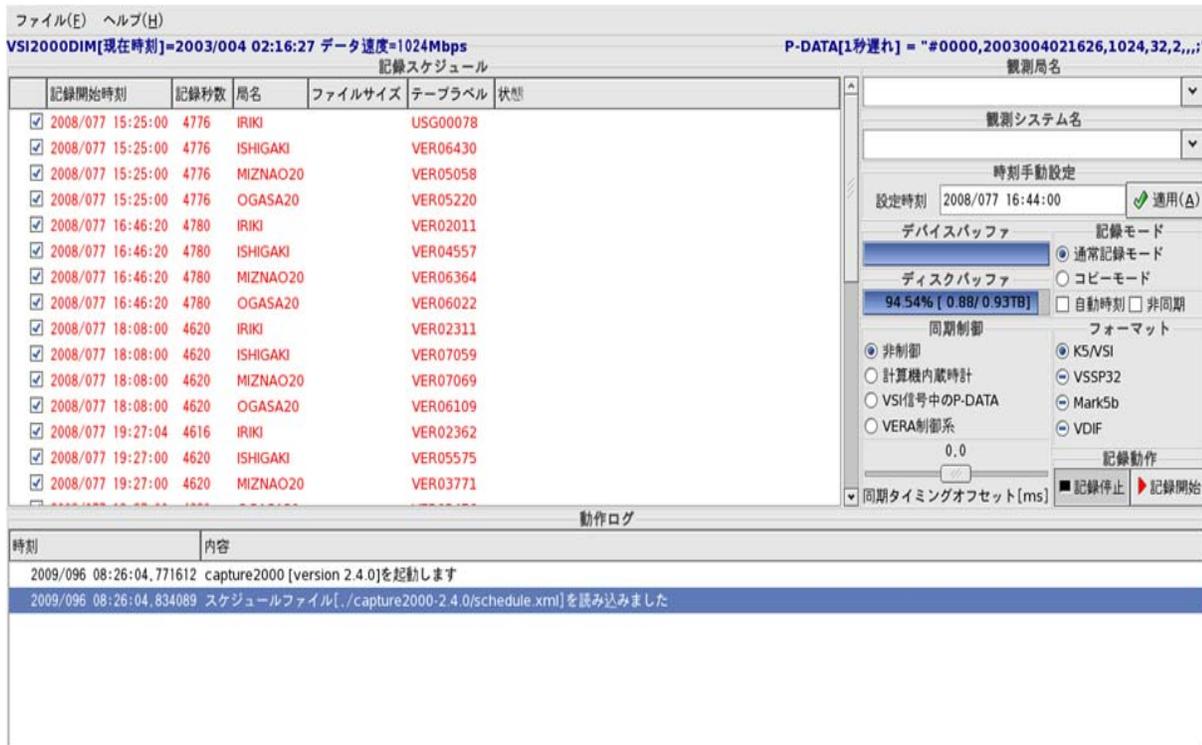
```
##### コマンドヘルプ #####
名前
    vsi2000_rec - VSI 信号をファイルへ記録します
書式
    vsi2000_rec --epoch='YYYY/DDD HH:MM:SS' --length=int
                --file=string <--help> <--version>
オプション
    --epoch        指定時刻 'YYYY/DDD HH:MM:SS' から記録を開始します
    --length       記録時間長 [秒] を指定します
    --file         記録ファイル名を設定します
    --verbose      記録中に詳細を表示します
    --help         この画面を表示します
    --version      このソフトのバージョンとコンパイルされた日時を表示します
#####
```

4.1.4 capture2000

数多くの記録を実行する場合には、XML形式のスケジュールファイルを使用することで GUI ベースのスケジュール記録ソフト capture2000¹¹が使用できます¹²。このソフトはスケジュールファイル中の複数の schedule⇒capture で記述された記録を逐次的に行ないます。以下に記録のみを目的としたスケジュールファイルの例を記載します¹³。

```
<?xml version='1.0' encoding='UTF-8' ?>
<schedule>
  <station key='R'><name>IRIKI</name> ... </station>
  <capture><epoch>2008/077 15:25:00</epoch><length>4776</length><station>IRIKI</station><label>USG00078</label></capture>
  <capture><epoch>2008/077 16:46:20</epoch><length>4780</length><station>IRIKI</station><label>VER02011</label></capture>
</schedule>
```

通常の観測では観測ファイルの保存場所をオプション”-raw-file”で指定し、さらに VERA 再生系からのテープコピーの場合には VSI サーバー、DMS24 サーバー、DIR2000 サーバーの IP アドレスとポート番号、用いているデッキ情報 BIN1—BIN2 を追加して capture2000 を起動します。capture2000 を起動後に上段のメニューバーのファイルからスケジュールファイルを選択します。ファイル選択後、通常の記録観測時には適切な測局名と観測システム名を選択します。テープコピー時にはここは空欄のままとします。通常の記録観測時には同期制御を ADS1000 互換機の場合は P-DATA、ntp で時計が制御されている場合は計算機内蔵時計を選択します。どちらも使用出来ない場合には、ここで非制御を選択し、時刻手動設定でマニュアルで時刻を合わせます¹⁴。VERA からのテープコピーではここで VERA 制御系を選択します。VERA 制御系を使用する場合には、TCP 通信を行なう必要がありますので、起動時にオプション VERA を指定してアドレスとポート番号を設定する必要があります。同期処理を VERA 制御系にした場合には、VSI 中の 1PPS 信号と TCP 経由で得られる時計の繰り上がりタイミングのずれを補正するため、同期タイミングオフセットを適宜指定することが出来ます¹⁵。記録モードは通常の観測時には通常記録モードを選択し、テープコピーの場合にはコピーモードを選択します。記録モードの下段の自動時刻と非同期はデバッグ用コマンドで通常は非選択状態で使用します¹⁶全ての設定後、記録動作の記録開始を押して記録動作を開始します。



¹¹ このソフトは /dev/mem にアクセスするため root での起動が必要です
¹² 記録・相関スケジュール参照
¹³ label 情報は VERA 再生系からのテープコピー時のみ使用します
¹⁴ 正午に時計を合わせる場合には”西暦/通日 12:00:00”と代入して、12:00:00+0.5 秒あたりで適用ボタンを押します
¹⁵ VSI 信号の 1PPS を検出したのち、250+オフセット [ミリ秒] 後に VERA 制御系に時計を读出しに行きます
¹⁶ 自動時刻は各スキャン開始時に時計を開始 5 秒前に再設定します。非同期は K5/VSI の時刻を無視して記録を行ないます

```

##### コマンドヘルプ #####
名前
    capture2000 - スケジュールに従ってファイルへ記録します
書式
    capture2000 --schedule=string --raw-file=string
                --VERA=string [--help] [--version]
オプション
    --schedule      スケジュールファイルを起動時に指定します
    --raw-file      記録先のディレクトリ名を指定します
    --VERA          VSI サーバに接続します [string=address:port]
    --DMS24         DMS24 サーバに接続します [string=address:port,bin]
    --DIR2000      DIR2000 サーバに接続します [string=address:port]
    --debug         デバッグ情報を表示します
    --help          この画面を表示します
    --version       バージョンとコンパイルされた日時を表示します
#####

```

4.1.5 gico3 (旧実行名:gico_corr)

ソフトウェア相関器 gico3 はスケジュールファイルを使用して一連の相関処理を実行する事が出来ます。相関処理に使用される観測ファイル、遅延情報ファイルと相関出力ファイルが保存されるディレクトリをオプションで指定します。遅延情報ファイルが指定されていない場合には gico3 内部で遅延計算されたものが使用されます。gico3 には多くのオプションが存在し、さらに今後追加されることがありますのでコマンドヘルプにより個々の機能を確認してください。

コマンドヘルプ

名前

gico3 - スケジュールファイルに従い相関処理を実行します

書式

```
gico3 --schedule=string --raw-file-string
      --geo-file=string --cor-file=string --multi=int
      --epoch='YYYY/DDD HH:MM:SS' --length=int
      --complex --command=string --remove=string
      [--help] [--version]
```

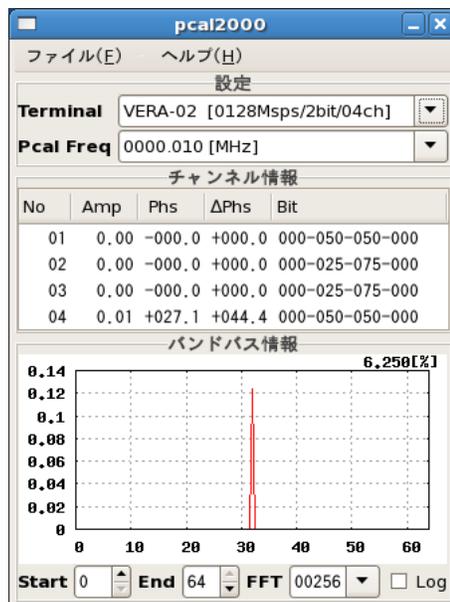
オプション

--schedule	GIC03 用の相関スケジュールファイルを指定します
--raw-file	観測ファイルの格納ディレクトリ名を指定します
--geo-file	遅延ファイルの格納ディレクトリ名を指定します
--cor-file	相関ファイルの格納ディレクトリ名を指定します
--epoch	[epoch:epoch+length-1] 間の相関処理のみを実行します
--length	[epoch:epoch+length-1] 間の相関処理のみを実行します
--multi	相関処理で使用するスレッド数を指定します
--boost	相関処理を間引きして実行速度を見掛け上向上させます
--complex	FFT 前にフリッジ回転処理をおこないます
--command	1 秒毎に指定したスクリプトファイルを起動します
--remove	指定した観測局を除いて相関処理を実行します
--verbose	相関処理中に詳細を表示します
--help	この画面を表示します
--version	ソフトのバージョンとコンパイルされた日時を表示します

#####

4.1.6 pcal2000

pcal2000 は ADS2000 用に開発された PCAL 信号モニターソフトにバンドパス表示機能を追加して、ADS1000 と VERA-1,2,4,7 に対応させたものでリアルタイムにバンドパスやビット分布が確認できます。ビット分布と PCAL は全データを使用し、バンドパスは全データ 16 分の 1 を計算に使用します。



コマンドヘルプ

名前

pcal2000 - ADSx000 の PCAL 情報等表示します

書式

pcal2000 [--help] [--version]

オプション

--help この画面を表示します

--version バージョンとコンパイルされた日時を表示します

#####

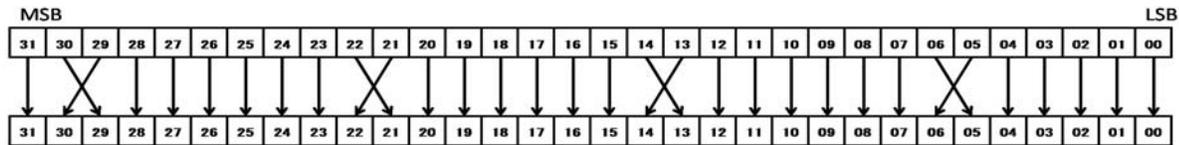
4.2 相関スケジュールの作成

4.2.1 異なるターミナル間での相関処理スケジュールファイル

GICO3 での相関処理は shuffle 機能により 32 ビット内の並びを自由に設定することが出来ます。この並び変えは変換前の 00 から 31 までのビットが変換後どのビットに対応するかを schedule⇒terminal⇒shuffle に指定します。GICO3 で正常に相関処理するためには変換後のビット並びは付録にあるデータの並びに合っていることが必要です。例えば 32 ビットの第 05 ビットと第 06 ビット、第 13 ビットと第 14 ビット、第 21 ビットと第 22 ビット、第 29 ビットと第 30 ビットを入れ替えて相関処理を行いたい場合には、下記のように 32 要素を設定します。

```
<shuffle key="A">  
31,29,30,28,27,26,25,24,23,21,22,20,19,18,17,16,15,13,14,12,11,10,09,08,07,05,06,04,03,02,01,00  
</shuffle>
```

記録データのビット並び



相関処理で使用するデータのビット並び

多ビットのデータをどのような数値にデコードするかは schedule⇒terminal⇒level で設定します。2 ビット量子化の場合に 2 進数で 00 から 11 までの 4 段階を -1.5, -0.5, +0.5, +1.5 として解釈させたい場合に以下のように記述します¹⁷。

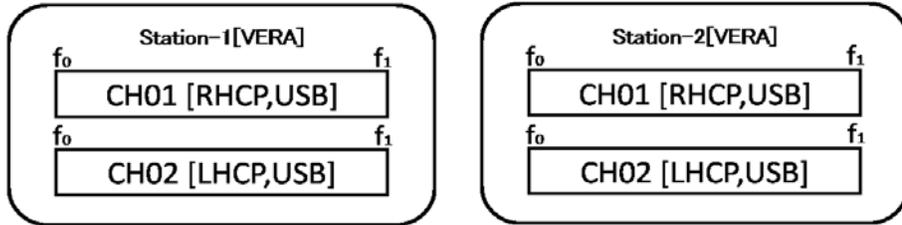
```
<level>-1.5,-0.5,+0.5,+1.5</level>
```

観測局毎に USB か LSB の違いがある場合には、schedule⇒process⇒stream⇒frequency に観測帯域内で DC よりの最下限値を記述し、schedule⇒process⇒stream⇒special⇒sideband に USB か LSB を記述することで混合相関処理が可能です。観測局ごとに観測周波数が若干異なる場合には、FFT 後に周波数領域で全体をシフトすることで周波数差を補正することが可能です。ここでシフトする量は schedule⇒process⇒stream⇒special⇒offset に記述します。さらに観測時のサンプリング周波数が異なる場合には、GICO3 内部で自動的に一番サンプリングレートが高い局を基準にして、それよりも小さいサンプリング局のデータは FFT 数を小さくして、周波数領域での分解能を等しくして相関処理が行われます。

¹⁷VERA 系の場合には -1.5, -0.5, +0.5, +1.5、ADSx000 系の場合には -1.5, +0.5, -0.5, +1.5 とする必要があります

4.2.2 偏波観測のための相関処理スケジュールファイル

相関処理で全ての偏波の組み合わせを計算する場合には、一つの観測局を仮想的に二局相当とみなした相関スケジュールファイルを作成します。ここでは、VERA-1 モードで Station1, Station2 の 2 局で 2 偏波観測した場合での相関スケジュールファイルの作成例を示します。



はじめに観測局を倍にするために局名に”-R”、”-L” を付け、それぞれの key を大文字と小文字で区別した合計 4 個の schedule⇒station を記述します。

```
<station key="A"><name>Station1-R</name><terminal>VERA-01</terminal></station>
<station key="a"><name>Station1-L</name><terminal>VERA-01</terminal></station>
<station key="B"><name>Station2-R</name><terminal>VERA-01</terminal></station>
<station key="b"><name>Station2-L</name><terminal>VERA-01</terminal></station>
```

GICO3 での相関処理では schedule⇒station⇒name で記述される観測局名から始まる観測ファイルを使用するため、デフォルトのままではこれらのファイルが存在しないため相関処理が失敗します。そこで相関処理に使用する観測ファイルを schedule⇒process⇒station⇒file で記述します。

```
<station key="A"><file>/mnt/raid/Station1_2003328013400.raw</file></station>
<station key="B"><file>/mnt/raid/Station2_2003328013400.raw</file></station>
<station key="a"><file>/mnt/raid/Station1_2003328013400.raw</file></station>
<station key="b"><file>/mnt/raid/Station2_2003328013400.raw</file></station>
```

それぞれの観測局で適切なチャンネル番号を使用するために、schedule⇒process⇒stream⇒special にチャンネル番号 1 と 2 を記述します。

```
<stream>
  <special key="A"><channel>01</channel></special>
  <special key="B"><channel>01</channel></special>
  <special key="a"><channel>02</channel></special>
  <special key="b"><channel>02</channel></special>
</stream>
```

以上の記述以外に汎用的な記述を追加し、実際に GICO3 で動作させるスケジュールファイルは以下のようになります。この例では一つのスキャン分しか記述されていませんが、複数の schedule⇒process を記述すれば数多くのスキャンに対応できます

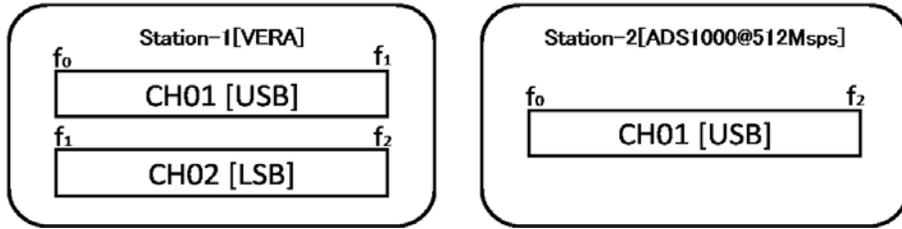
```
<?xml version="1.0" encoding="UTF-8" ?>
<schedule>
  <terminal name="VERA-01">
    <speed> 256000000</speed><channel> 2</channel><bit>2</bit><level>-1.5,-0.5,+0.5,+1.5</level>
  </terminal>
  <station key="A"><name>Station1-R</name><terminal>VERA-01</terminal></station>
  <station key="a"><name>Station1-L</name><terminal>VERA-01</terminal></station>
  <station key="B"><name>Station2-R</name><terminal>VERA-01</terminal></station>
  <station key="b"><name>Station2-L</name><terminal>VERA-01</terminal></station>

  <source name="NAME"><ra>00h00m00.000000</ra><dec>+00d00'00.00000</dec></source>

  <process>
    <start>2003/328 01:34:00</start><length>3960</length><object>Multi</object>
    <stations>AaBb</stations>
    <station key="A"><file>/mnt/raid/Station1_2003328013400.raw</file></station>
    <station key="B"><file>/mnt/raid/Station2_2003328013400.raw</file></station>
    <station key="a"><file>/mnt/raid/Station1_2003328013400.raw</file></station>
    <station key="b"><file>/mnt/raid/Station2_2003328013400.raw</file></station>
    <stream>
      <label>CH01</label><source>NAME</source><frequency>2222000000</frequency>
      <channel>01</channel><fft>2048</fft><output>1</output>
      <special key="A"><channel>01</channel></special>
      <special key="B"><channel>01</channel></special>
      <special key="a"><channel>02</channel></special>
      <special key="b"><channel>02</channel></special>
    </stream>
  </process>
</schedule>
```

4.2.3 VERA-1 モードと ADS1000[512Msps/2bit/01ch] の混合相関処理

下図のような周波数配置で VERA-1 モードと ADS1000[512Msps/2bit/01ch] の混合相関をする場合には、基本的な設定は VERA-1 モードと同じで 2 チャンネル分の相関処理スケジュールファイルを作成します。

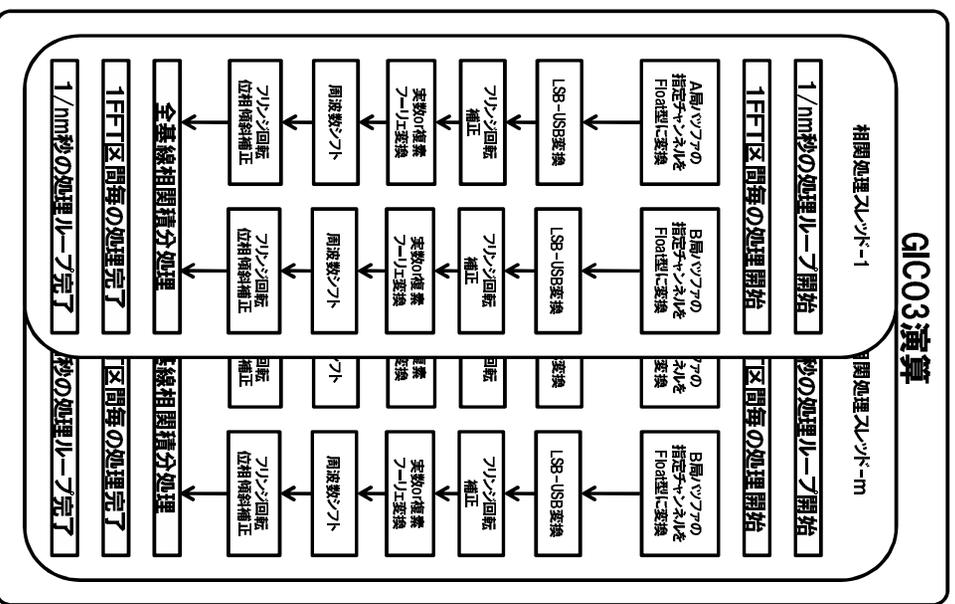
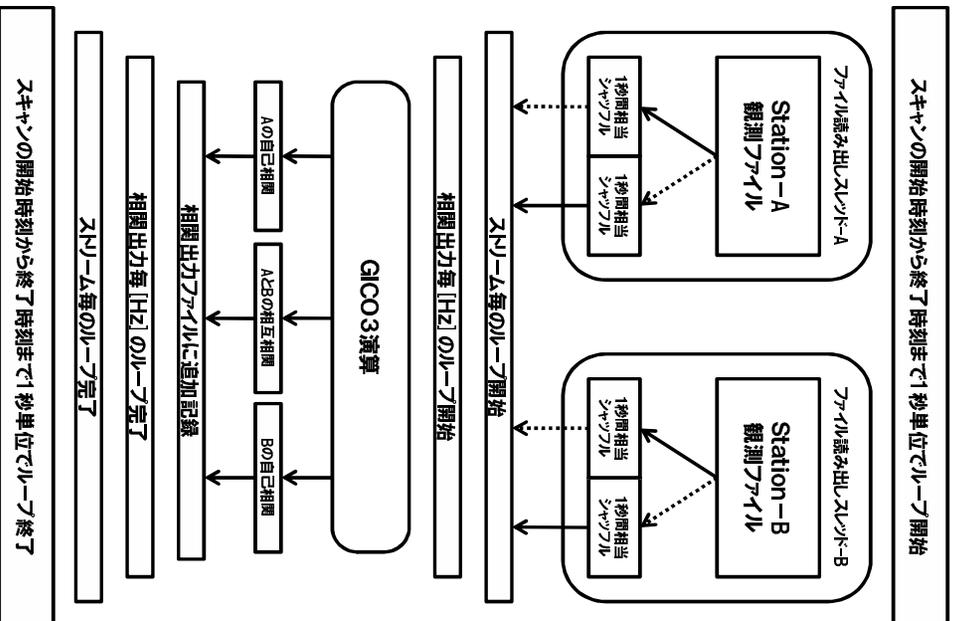


VERA 側の第一チャンネルと ADS1000 側の第一チャンネルは観測周波数の下限が同じなので、正しく terminal が定義されていれば、通常のスケジュールのままに相関処理が可能です。VERA 側の第二チャンネルと ADS1000 側の第一チャンネルはチャンネル番号と観測周波数が異なるので、`schedule⇒process⇒stream⇒special⇒channel`、`schedule⇒process⇒stream⇒special⇒offset`、でチャンネル番号と周波数オフセットを記述する必要があります。相関処理では VERA 局では第二チャンネル、ADS1000 局では第一チャンネルを使用するため、channel に 2 と 1 をそれぞれ記述します。GICO3 ではサンプリング速度が最も高い 512Msps 時に 2048 点 FFT が行われるため、FFT 後の周波数分解能は 0.25MHz となります。128MHz は 512 個相当のオフセット量となるため、ADS1000 局側の offset に -512 を記述します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<schedule>
  <terminal name="VERA-01">
    <speed> 256000000</speed><channel> 2</channel><bit>2</bit><level>-1.5,-0.5,+0.5,+1.5</level>
  </terminal>
  <terminal name="ADS1000">
    <speed> 512000000</speed><channel> 1</channel><bit>2</bit><level>-1.5,+0.5,-0.5,+1.5</level>
  </terminal>
  <station key="A"><name>Station-1</name><terminal>VERA-01</terminal></station>
  <station key="B"><name>Station-2</name><terminal>ADS1000</terminal></station>
  <source name="NAME"><ra>00h00m00.000000</ra><dec>+00d00'00.00000</dec></source>
  <process>
    <start>2003/328 01:34:00</start><length>3960</length><object>Multi</object>
    <stations>AB</stations>
    <stream>
      <label>CH01</label><source>NAME</source><frequency>2200000000</frequency>
      <channel>01</channel><fft>2048</fft><output>1</output>
    </stream>
    <stream>
      <label>CH02</label><source>NAME</source><frequency>2212800000</frequency>
      <channel>02</channel><fft>2048</fft><output>1</output>
      <special key="A"><channel>02</channel><offset>+0000</offset><sideband>LSB</sideband></special>
      <special key="B"><channel>01</channel><offset>-0512</offset><sideband>USB</sideband></special>
    </stream>
  </process>
</schedule>
```

4.3 GICO3の内部動作

FX型の多基線相関器として設計されているGICO3の内部動作について簡単に説明します。ここでは単純化のため2局分のデータを相関処理する場合について、次ページにその内部処理を示します。ソフトウェア相関器の処理は内部で1秒単位で実行されており、相関処理に必要な観測データを準備する複数の読み出しスレッドと、実際の演算を実行する複数のスレッドが並列に動作しています。読み出しスレッドは演算を実行中に次に必要になる1秒分のデータをファイルから読み出しており、必要があればここで読み出されたデータのシャッフルも行います。ソフトウェア相関処理では1秒間の処理をストリームとよばれる単位に分解して処理を行います。1個のストリームは1個のチャンネル番号を記述することができます。さらに、各ストリームは相関処理スケジュールの `schedule⇒process⇒stream⇒output` で記述される相関出力速度 ($n[\text{Hz}]$) 毎に細分化され、プロセッサのコア数と等しい m 個のスレッドからなる GICO3 演算コアにより分散処理が行われます。GICO3 演算コア内部では1つのスレッドが $1/mn$ 秒相当の処理を独立して行い、1FFT 区間に相当する時間単位で処理が進められます。GICO3 演算コアははじめに観測データから指定されたチャンネル番号を選択して単精度型の浮動小数点に変換します。この時使用されるチャンネル番号は `schedule⇒process⇒stream⇒channel` に記述されているものを使用しますが、観測局毎に異なる選択をする場合には `schedule⇒process⇒stream⇒special⇒channel` で個別に記述します。GICO3 での相関処理はすべて USB で計算されるため、LSB の場合にはここで USB に変換されます。GICO3 での相関処理では処理速度を向上させるため実数 FFT が使用されますが、基線長が長く観測周波数が高い場合には問題を生じる場合がありますので、相関処理の引数に `-complex` が指定されている場合には、FFT の前段でフリッジ回転補正が行われた後、複素 FFT が使用されます。FFT では全局同じ FFT 数の変換が実行されますが、異なるサンプリング速度間の混合相関処理の場合には、もっともサンプリング速度が大きい観測局のデータを `schedule⇒process⇒stream⇒fft` で記述された FFT 数でフーリエ変換を行い、それよりサンプリング速度が小さい観測局のデータは周波数分解能が等しくなるように小さい FFT 数でフーリエ変換が行われます。フーリエ変換後にそれぞれの周波数分解能の単位で `schedule⇒process⇒stream⇒special⇒offset` で信号をシフトさせた後、群遅延相当の位相傾斜の補正と引数 `complex` が指定されていない場合はここでフリッジ回転補正が実行されます。サンプリング速度の違いや周波数シフトにより観測信号の無い周波数成分にはゼロが代入された後、全組み合わせでの相関が行われ 1FFT 区間の処理が完了します。全ての GICO 演算コアで $1/mn$ 秒分の処理が完了した後に全スレッド分の処理結果が一つに結合され、最終的に1つのファイルとして出力されます。



GIC03 の内部動作

4.4 各ファイルの命名規則

capture2000 及び gico3_corr 等で取扱う各種ファイルの名前は以下の命名規則により決定されています。スケジュールファイル中に個々のファイル名を明示的に記述していなければ、暗黙的にこの規則が適応されます。但し、VERA バックアップ関連処理では途中の処理を簡便に行なうために、処理の過程において一部拡張型を使用する事があります。

表 2: 各ファイルの命名規則 (基本型)

ファイルの種類	命名規則	例
観測データファイル	局名+"_" +開始時刻 [YYYYDDDDHHMMSS]+".raw"	IRIKI_2003328013400.raw
遅延情報ファイル	局名+"_" +開始時刻 [YYYYDDDDHHMMSS]+ラベル名+".geo"	IRIKI_2003328013400_CH01.geo
遅延情報ファイル	局名+"_" +開始時刻 [YYYYDDDDHHMMSS]+天体名+".geo"	IRIKI_2003328013400_3C345.geo
相関結果ファイル	局名+"_" +局名+"_" +開始時刻 [YYYYDDDDHHMMSS]+ラベル名+".cor"	IRIKI_IRIKI_2003328013400_CH01.cor

表 3: 各ファイルの命名規則 (拡張型)

ファイルの種類	命名規則	例
遅延情報ファイル	局名+"_" +YYYYDDDDHHMMSS"+ラベル名+".geo"	IRIKI_YYYYDDDDHHMMSS_CH01.geo
遅延情報ファイル	局名+"_" +YYYYDDDDHHMMSS"+天体名+".geo"	IRIKI_YYYYDDDDHHMMSS_3C345.geo
相関結果ファイル	局名+"_" +局名+"_" +YYYYDDDDHHMMSS"+ラベル名+".cor"	IRIKI_IRIKI_YYYYDDDDHHMMSS_CH01.cor

4.5 遅延情報ファイル構造

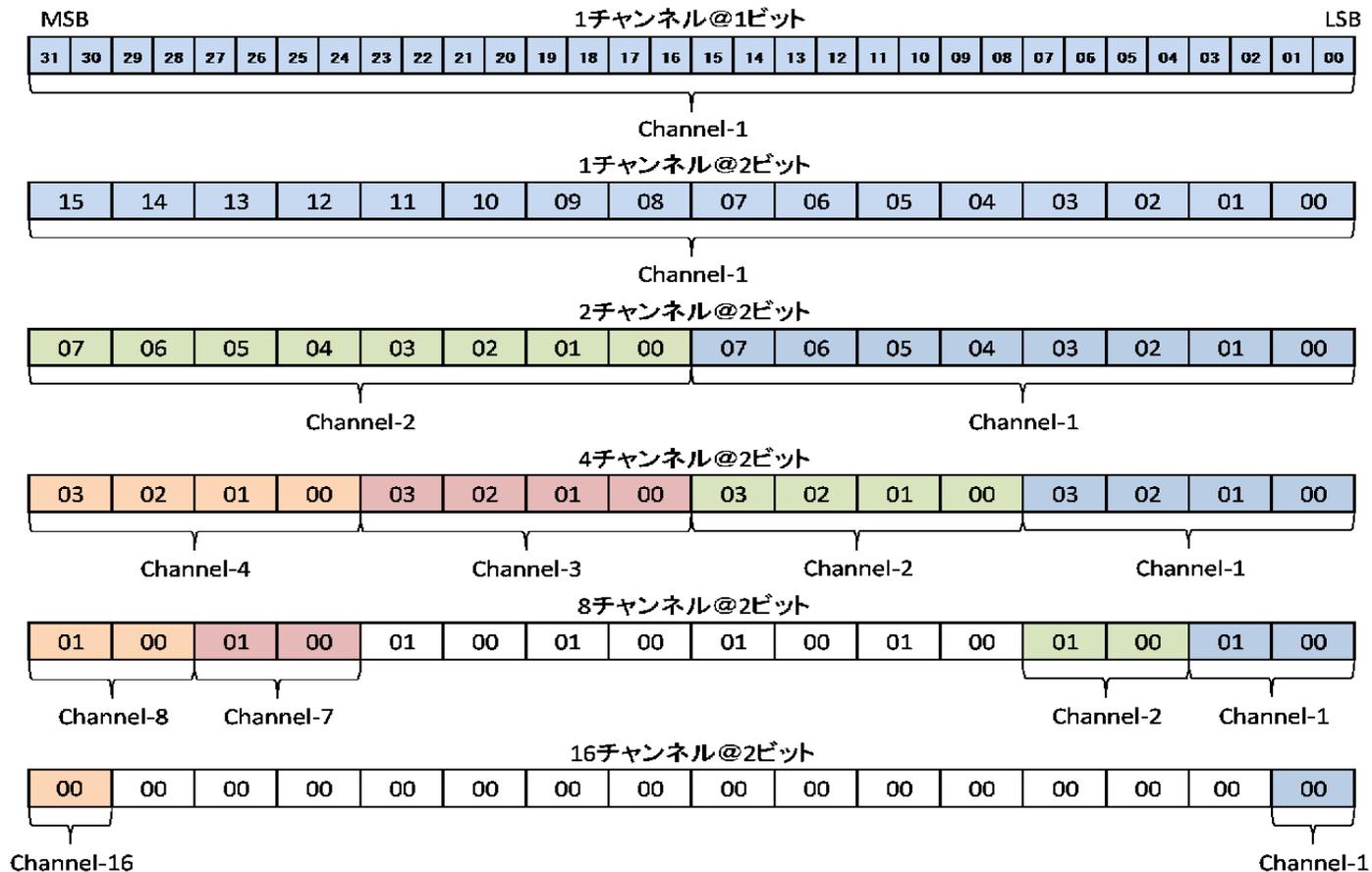
GICO3 で使用される遅延情報ファイルは時刻情報と遅延情報から構成されており、時刻情報は'1970/001 00:00:00'からの経過時刻の秒数を格納する 4 バイト長の *time_t* 型とナノ秒を格納する 4 バイト長の整数型で構成されています。ここでの遅延量は 4 次の微分係数までを含む倍精度浮動小数点型として格納されています。

表 4: 遅延情報ファイルのフォーマット (リトルエンディアン型/IEEE 754 型準拠)

オフセット	00	01	02	03	04	05	06	07
0x00000000	時刻 [秒 : time_t]				時刻 [ナノ秒 : int]			
0x00000008	遅延量 0 階微分 [double]							
0x00000010	遅延量 1 階微分 [double]							
0x00000018	遅延量 2 階微分 [double]							
0x00000020	遅延量 3 階微分 [double]							
0x00000028	遅延量 4 階微分 [double]							
0x00000030	時刻 [秒 : time_t]				時刻 [ナノ秒 : int]			
0x00000038	遅延量 0 階微分 [double]							
0x00000040	遅延量 1 階微分 [double]							
0x00000048	遅延量 2 階微分 [double]							
0x00000050	遅延量 3 階微分 [double]							
0x00000058	遅延量 4 階微分 [double]							
.....	.							

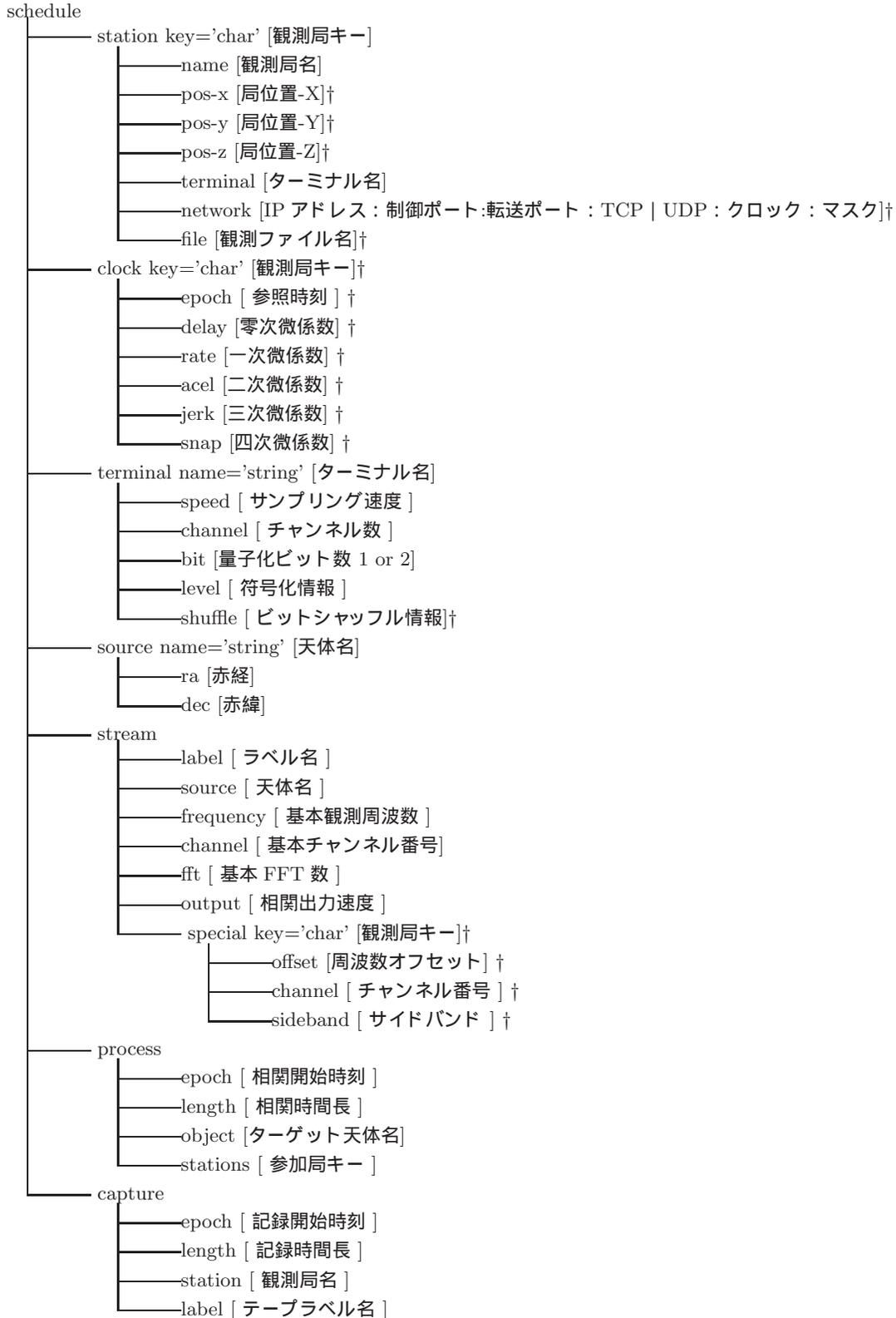
4.6 GICO3の相関処理でデータの並び

多チャンネルのデータ进行处理する場合には、観測ファイル中の各サンプルの並びは下図のようにになっている必要があります。この並びとは異なる場合には shuffle 機能を使用してこの並びに一致させる必要があります。なお、各サンプルがどの数値に変換されるかは、schedule⇒terminal⇒levelにより決定されます。



4.7 記録・関連スケジュール [構造]

capture2000 と gico3 のスケジュールファイルは以下のような構造を持つ共通ファイルフォーマットを採用しています。ここで † 付きのものは省略可能となっており、省略された場合は未使用又はデフォルト値又はゼロが代入されます。また、process 内に station,clock,terminal,source,stream を記述することにより、process 毎に異なる条件で相関処理が可能です。



4.8 記録・関連スケジュール [サンプル]

```
<?xml version='1.0' encoding='UTF-8' ?>
<schedule>
  <terminal name='VERA'><speed> 32000000</speed><channel>02</channel><bit>2</bit><level>-1.5,-0.5,+0.5,+1.5</level></terminal>
  <station key='R'>
    <name>IRIKI </name><pos-x>-3521719.568700</pos-x><pos-y>+4132174.752800</pos-y><pos-z>+3336994.325500</pos-z></station>
  <station key='I'>
    <name>ISHIGAKI</name><pos-x>-3263994.648300</pos-x><pos-y>+4808056.355800</pos-y><pos-z>+2619949.394400</pos-z></station>
  <station key='M'>
    <name>MIZNA020</name><pos-x>-3857241.855200</pos-x><pos-y>+3108784.850900</pos-y><pos-z>+4003900.585800</pos-z></station>
  <station key='G'>
    <name>OGASA20 </name><pos-x>-4491068.894000</pos-x><pos-y>+3481544.829500</pos-y><pos-z>+2887399.622700</pos-z></station>
  <clock key='R'><epoch>2008/077 15:10:00</epoch><delay>+4.9000e-06</delay><rate>+2.0900e-12</rate></clock>
  <clock key='I'><epoch>2008/077 15:10:00</epoch><delay>+4.6000e-07</delay><rate>+1.7000e-12</rate></clock>
  <clock key='M'><epoch>2008/077 15:10:00</epoch><delay>-5.0700e-06</delay><rate>+0.0000e+00</rate></clock>
  <clock key='G'><epoch>2008/077 15:10:00</epoch><delay>+1.1080e-05</delay><rate>+2.7700e-12</rate></clock>
  <capture><epoch>2008/077 15:25:00</epoch><length>4776</length><station> IRIKI</station><label>USG00078</label></capture>
  <capture><epoch>2008/077 15:25:00</epoch><length>4776</length><station>ISHIGAKI</station><label>VER06430</label></capture>
  <capture><epoch>2008/077 15:25:00</epoch><length>4776</length><station>MIZNA020</station><label>VER05058</label></capture>
  <capture><epoch>2008/077 15:25:00</epoch><length>4776</length><station> OGASA20</station><label>VER05220</label></capture>
  <process>
    <source name='DUM'><ra>00h00m00.000000</ra><dec>+90d00'00.000000 </dec></source>
    <source name='1641+399'><ra>16h42m58.809950</ra><dec>+39d48'36.993990 </dec></source>
    <epoch>2008/077 15:25:00</epoch><length>0060</length><object>Multi</object><stations>RIMG</stations>
    <stream>
      <label>CH01</label><source> DUM</source><frequency>+2222600000</frequency><channel>01</channel>
      <fft>002048</fft><output>001</output>
    </stream>
    <stream>
      <label>CH02</label><source>1641+399</source><frequency>+2200200000</frequency><channel>02</channel>
      <fft>002048</fft><output>001</output>
    </stream>
    <station key='R'><file> IRIKI_2008077152500.raw</file></station>
    <station key='I'><file>ISHIGAKI_2008077152500.raw</file></station>
    <station key='M'><file>MIZNA020_2008077152500.raw</file></station>
    <station key='G'><file> OGASA20_2008077152500.raw</file></station>
  </process>
</schedule>
```


表 6: 付録 相関結果ファイルの現フォーマット (Version 1.1)

Address	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
Header Region																
0x000-0x00f	Magic Word		0x3ea2f983		Header Version		0x01010000		Software Version		int		Sampling Speed		int	
0x010-0x01f	Observed Sky Frequency		double		Station-1 Name		char × 16		FFT-Point		int		Number of Sector		int	
0x020-0x02f	Station-1 Position-X		double		Station-1 Position-Y		double		Free-Area		default : 0x00000000					
0x040-0x04f	Station-1 Position-Z		double		Station-2 Name		char × 16		Station-2 Position-X		double		Station-2 Position-Y		double	
0x050-0x05f	Station-2 Position-Z		double		Source Name		char × 16		Free-Area		default : 0x00000000					
0x060-0x06f	Source Position-Ra		double		Station-1 Clock Epoch NanoSec		int		Source Position-Dec		double		Station-1 Clock Delay		double	
0x070-0x07f	Station-1 Clock Rate		double		Station-1 Clock Jerk		double		Station-1 Clock Acel		double		Station-1 Clock Acel		double	
0x080-0x08f	Station-2 Clock Rate		double		Station-2 Clock Jerk		double		Station-2 Clock Acel		double		Station-2 Clock Acel		double	
0x090-0x09f	Station-2 Clock Epoch Sec		time_t		Station-2 Clock Epoch NanoSec		int		Station-2 Clock Jerk		double		Station-2 Clock Acel		double	
0x0a0-0x0af	Station-1 Model Rate		double		Station-1 Model Jerk		double		Station-1 Model Acel		double		Station-1 Model Acel		double	
0x0b0-0x0bf	Station-2 Model Rate		double		Station-2 Model Jerk		double		Station-2 Model Acel		double		Station-2 Model Acel		double	
0x0c0-0x0cf	Effective Integration Length		float		Real[0]		float		Imag[0]		float		Real[1]		float	
0x0d0-0x0df	Imag[0]		float		Imag[1]		float		Real[1]		float		Imag[1]		float	
0x0e0-0x0ef	Real[0]		float		Imag[0]		float		Real[0]		float		Imag[0]		float	
0x0f0-0x0ff	Real[1]		float		Imag[1]		float		Real[1]		float		Imag[1]		float	
First Sector																
0x100-0x10f	Correlation Start Sec		time_t		Correlation Start NanoSec		int		Correlation Stop Sec		time_t		Correlation Stop NanoSec		int	
0x110-0x11f	Station-1 Model Epoch Sec		time_t		Station-1 Model Epoch NanoSec		int		Station-1 Model Delay		double		Station-1 Model Acel		double	
0x120-0x12f	Station-1 Model Rate		double		Station-1 Model Jerk		double		Station-1 Model Acel		double		Station-1 Model Acel		double	
0x130-0x13f	Station-2 Model Rate		double		Station-2 Model Jerk		double		Station-2 Model Acel		double		Station-2 Model Acel		double	
0x140-0x14f	Station-1 Model Epoch Sec		time_t		Station-1 Model Epoch NanoSec		int		Station-1 Model Delay		double		Station-1 Model Acel		double	
0x150-0x15f	Station-2 Model Epoch Sec		time_t		Station-2 Model Epoch NanoSec		int		Station-2 Model Delay		double		Station-2 Model Acel		double	
0x160-0x16f	Station-1 Model Rate		double		Station-1 Model Jerk		double		Station-1 Model Acel		double		Station-1 Model Acel		double	
0x170-0x17f	Station-2 Model Rate		double		Station-2 Model Jerk		double		Station-2 Model Acel		double		Station-2 Model Acel		double	
0x180-0x18f	Effective Integration Length		float		Real[0]		float		Imag[0]		float		Real[1]		float	
0x190-0x19f	Imag[0]		float		Imag[1]		float		Real[1]		float		Imag[1]		float	
0x1a0-0x1af	Real[0]		float		Imag[0]		float		Real[0]		float		Imag[0]		float	
0x1b0-0x1bf	Real[1]		float		Imag[1]		float		Real[1]		float		Imag[1]		float	
0x1c0-0x1cf	Imag[0]		float		Imag[1]		float		Real[0]		float		Imag[0]		float	
0x1d0-0x1df	Imag[1]		float		Real[0]		float		Imag[1]		float		Real[1]		float	
0x1e0-0x1ef	Real[0]		float		Imag[0]		float		Real[0]		float		Imag[0]		float	
0x1f0-0x1ff	Real[1]		float		Imag[1]		float		Real[1]		float		Imag[1]		float	
Second Sector																
0x200-0x20f	Correlation Start Sec		time_t		Correlation Start NanoSec		int		Correlation Stop Sec		time_t		Correlation Stop NanoSec		int	
0x210-0x21f	Station-1 Geometry-Sec		time_t		Station-1 Geometry-NanoSec		int		Station-1 Geometry-Delay		double		Station-1 Geometry-Delay		double	
0x220-0x22f	Station-1 Geometry Rate		double		Station-1 Geometry Jerk		double		Station-1 Geometry Acel		double		Station-1 Geometry Acel		double	
0x230-0x23f	Station-2 Geometry Rate		double		Station-2 Geometry Jerk		double		Station-2 Geometry Acel		double		Station-2 Geometry Acel		double	
0x240-0x24f	Effective Integration Length		float		Real[0]		float		Imag[0]		float		Real[1]		float	
0x250-0x25f	Imag[0]		float		Imag[1]		float		Real[1]		float		Imag[1]		float	
0x260-0x26f	Real[0]		float		Imag[0]		float		Real[0]		float		Imag[0]		float	
0x270-0x27f	Real[1]		float		Imag[1]		float		Real[1]		float		Imag[1]		float	