

TR-I-0037

Speech Workbench Ver.3

音声処理ワークベンチVer.3

Koichi Murayama, Takeshi Kawabata

村山浩一、川端豪

1988.8

概要

音声処理ワークベンチVersion.3の操作方法及び、仕様について報告する。音声処理において、波形データを編集・加工できることはとても重要なことである。そこでワークベンチVer.3では、Ver.2の持っていた機能は全て含み、加えてユーザが音声を実際に聞きながら編集・加工出来る波形エディタの機能を持つ。また高速かつ高精度な画面表示をワークステーション上に実現した。

第一部 使用の手引

目 次	ページ
1. 概要	1
1. 1 起動方法	2
1. 2 終了方法	2
2. 波形エディタ	
2. 1 ウィンドウ説明	
2. 1. 1 ウィンドウ名	3
2. 1. 2 ウィンドウオペレーション	4
2. 2 D/A出力	6
2. 3 A/D入力	6
2. 4 FILE操作	7
2. 4. 1 LOAD	
2. 4. 2 SAVE	
2. 4. 3 PATH	
2. 4. 4 LABEL	
2. 5 PARAMETER	8
2. 5. 1 sampling_rate	
2. 5. 2 AutoCorrelation	
2. 5. 3 RUNNING_SPECTRUM	
2. 5. 4 FET_SPECTRUM_SLICE	

目	次	ページ
2. 6	AREA	10
2. 6. 1	RESET	
2. 6. 2	LEFT	
2. 6. 3	RIGHT	
2. 7	ZOOM	11
2. 7. 1	ZOOM_UP	
2. 7. 2	ZOOM_DOWN	
2. 7. 3	STANDARD	
2. 8	EDIT	12
2. 8. 1	COPY	
2. 8. 2	REMOVE	
2. 8. 3	INSERT	
2. 8. 4	SILENCE	
2. 8. 5	MIX	
2. 8. 6	GAIN	
2. 8. 7	UNDO	

- 3. アナライザ
 - 3. 1 MakeSpectrum 13
 - 3. 2 SPECTROGRAM 13
 - 3. 3 MakeAutoCorrelation 13
 - 3. 4 LOG_POWER 13
 - 3. 5 Running_Spectrum 14
 - 3. 6 FET_SPECTRUM_SLICE 14
 - 3. 7 LPC_SPECTRUM_SLICE 14

- 4. その他の機能 17
 - 4. 1 LABEL
 - 4. 2 REDRAW
 - 4. 3 GENERATER
 - 4. 4 LOCATION
 - 4. 5 ICONS
 - 4. 6 CLEAR
 - 4. 7 WINDOW

- Appendix 19
 - Appendix_A セットアップの仕方
 - Appendix_B ラインエディット機能について

1. 概要

ワークベンチVer3.0は、大きく分けて、音声データを編集・加工できる波形エディタと、分析及び解析等を行うアナライザから構成されている。

ワークベンチVer2で持っていた機能(アナライザ)は全部含み、しかもその全てをXウィンドウ・システムで書き直してある。

この為、以下のような特徴を持つ。

- ・画面破壊時の復旧が早い。
- ・全体的にグラフィック出力が高速である。
- ・音声の入出力及び編集・加工がマウスにより容易に行える。
- ・同時に3種類の音声データを編集できる。
- ・各音声データは最大6秒間まで持つことができる。
- ・分析結果を表示しているウィンドウをresizeすることができる。
- ・不必要な表示はICON化できる。

1.1 起動方法

以下のコマンドを入力することにより、起動させることができる。

```
% wedt
```

なお、ファイル出力時のパス指定を引数で指定することができる。

example)

```
% wedt /data1 /data2 /data3
```

引数はそれぞれ、3つの編集用画面に相当する。又、引数の数により起動時の編集用画面の数が決定する。

ただし、第2、第3引数を省略した場合は、1つ前の引数と同じになる。

example)

```
% wedt /data1 /data2
```

```
      ||  
% wedt /data1 /data2 /data2
```

```
% wedt /data1
```

```
      ||  
% wedt /data1 /data1 /data1
```

1.2 終了方法

波形編集用ウィンドウにマウスカーソルを入れ、キーボードのSHIFTボタンとマウスの中央ボタンを同時にクリックすることにより、終了する。

2. 波形エディタ

2.1 ウィンドウ説明

2.1.1 ウィンドウ名

波形エディタは、図2.1.1のようなウィンドウが最大3つ作成できる。
以下に各ウィンドウ中の名称及び機能を説明する。

(1) ファンクション・メニュー

図2.1.1のように、8個のメインメニューから構成されている。マウスをクリックすることにより、サブメニューが表示される。

(2) 表示ゲイン・インジケータ

表示されている波形のゲイン(振幅)を、マウスクリックにより変更できる。
ただし、この操作では実データには影響しない。

(実データの振幅を変更したいときは、2.8.5を参照のこと)
デフォルトでは、波形表示ウィンドウの最下段が-32768、最上段が32768で表示される。

(3) 編集ファイル名

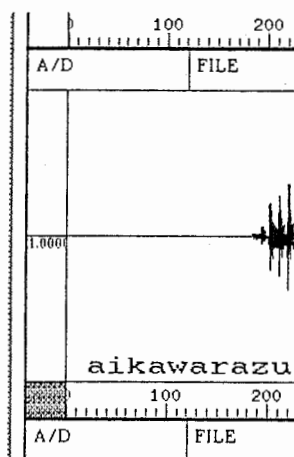
音声データをディスクから読み込んだ場合、及び書き込んだ時に、そのファイル名を表示する。

(4) セーブ・フラグ

音声データを編集した後に、セーブされていない場合、ユーザに注意を促すものである。(図2.1.2)

このウィンドウ内でマウスクリックすると、画面全体がアイコンになる。

図2.1.2



(5)時間スケール

現在表示されている音声データの時間軸を表示する。(単位ms)

このスケール上でマウスをクリックすることにより、波形表示を左右に移動することができる。

例えば1200msのところでもマウスをクリックすると、画面の中央に1200msの位置が移動する。

2.1.2 ウィンドウ オペレーション

(1)波形編集用ウィンドウの削除

波形出力ウィンドウにマウスカーソルを入れ、キーボードのCTRLボタンとマウスの中央ボタンを同時にクリックすると、“Sure? (Y/N)” のメッセージが出力される(出力されている波形が全て0の場合、無条件でそのウィンドウが削除される)ので、“Y” 又は“y” を入力するとウィンドウが削除される。また、そのウィンドウに関するアナライズもKillされる。

尚、波形編集用ウィンドウが一つの場合、このオペレーションは無視される。

(2)波形編集用ウィンドウの追加

4. その他の機能、4.7 WINDOW、参照。

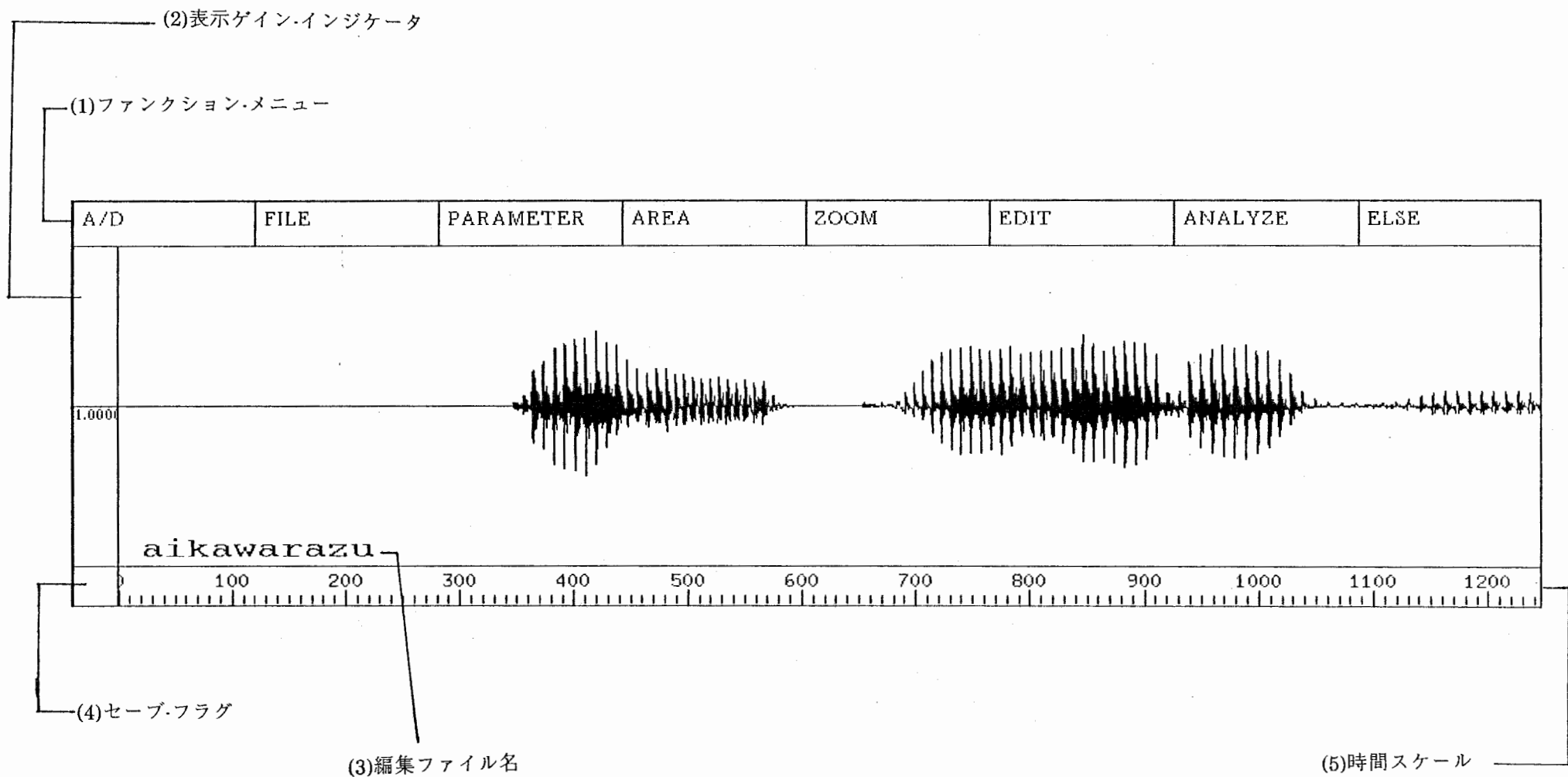


図2.1.1

2.2 D/A出力

マウス・カーソルがスピーカの形をしている時にクリックすると、選択された画面の音声データが、D/A出力される。

ただし、現在画面上に表示されている部分のみが出力される。(ズーム・アップ時等) またエリア指定時には、その指定された範囲が出力される。

音声の反転出力も可能である。(2.6AREA参照)

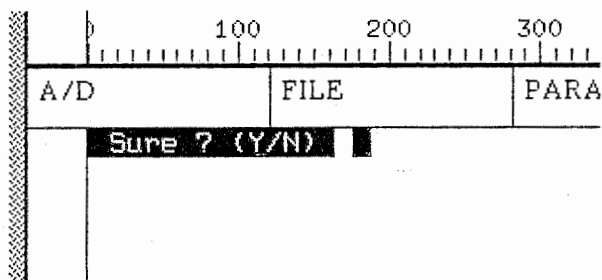
2.3 A/D入力

ファンクション・メニューによりA/Dをマウスクリックすると、A/D入力待ち状態になる。

音声区間は自動的に検知するようになっている。

尚、対象となるウィンドウに編集中の音声データがある(つまり、まだセーブされていない)場合は、確認の入力を要求してくるので、Yまたはyを入力するとA/D入力可能となる。(図2.3.1) ただし、バッファ中のデータは失われる。

図2.3.1



2.4 FILE操作

ファンクションメニューによりFILEをマウスクリックすると、サブメニューが表示される。(図2.4.1)

2.4.1 LOAD

ディスク上より、音声データをバッファに取り込む。ファイル名の指定はメニューにより行う。ただし、対象となるディレクトリは先にPATHにより指定しておかなければならない。

尚、対象となるウィンドウに編集中の音声データがある(つまり、まだセーブされていない)場合は、確認の入力を要求してくるので、Yまたはyを入力すると、新規ファイル読み込み可能となる。ただし、バッファ中のデータは失われる。

2.4.2 SAVE

音声データをディスク上にセーブする。ただし、対象となるディレクトリは先にPATHにより指定しておかなければならない。

2.4.3 PATH

ディスク操作のためのPATHを設定する。

LOAD-SAVE及びLABEL操作を行う前に、指定しなければならない。

デフォルトはカレントディレクトリである。

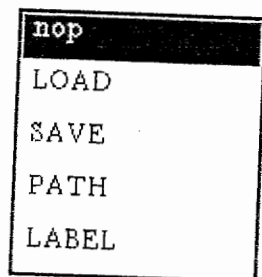
絶対指定及び相対指定いずれも可能である。

2.4.4 LABEL

ラベルファイルが存在する場合に、そのファイル名を入力する。

ただし、対象となるディレクトリは先にPATHにより指定しておかなければならない。ラベルの表示は、4.1LABELを参照。

図2.4.1



2.5 PARAMETER

音声データに関する情報を入力する。(図2.5.1)

2.5.1 sampling_rate

サンプリング・レートを入力する。(図2.5.1.1)

デフォルトは、12.0KHzである。

ここで入力しても、REDRAW(4.2参照)するまでは、画面表示はそのままである。

2.5.2 AutoCorrelation

相関係数計算を行うためのパラメータを設定する。

各パラメータの入力は、図2.5.2のようなサブ・メニューにより行う。

2.5.2.1 window

hamming,hanning,及びrectのいずれかを選択する。(図2.5.2.1) デフォルトはhamming。

2.5.2.2 window-length

分析ウィンドウ長を入力する。デフォルトは256。(図2.5.2.2)

2.5.2.3 frame-shift

フレーム・シフト数を入力する。デフォルトは64。(図2.5.2.3)

2.5.3 RUNNING_SPECTRUM

2.5.3.1 lpc

LPC次数を入力する。デフォルトは13。(図2.5.3.1)

2.5.4 FFT_SPECTRUM_SLICE

2.5.4.1 fft_length

FFT長を入力する。デフォルトは512。(図2.5.4.1)

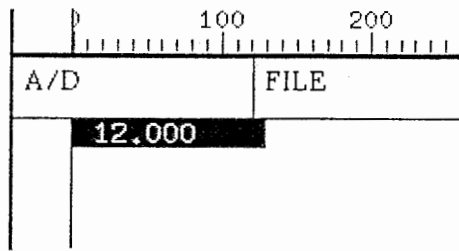
☒2.5.1

```

nop
sampling rate
AutoCorrelation
RUNNING<SPEC
FFT<SPECTRUM

```

☒2.5.1.1



☒2.5.2

```

nop
window
window-length
frame-shift

```

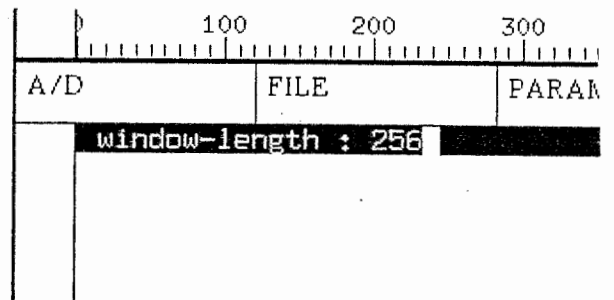
☒2.5.2.1

```

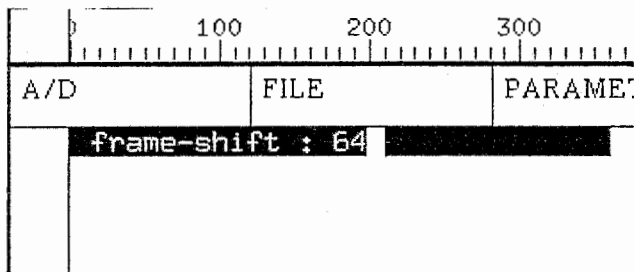
hamming
hanning
rect

```

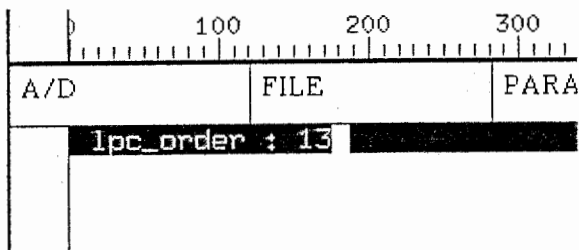
☒2.5.2.2



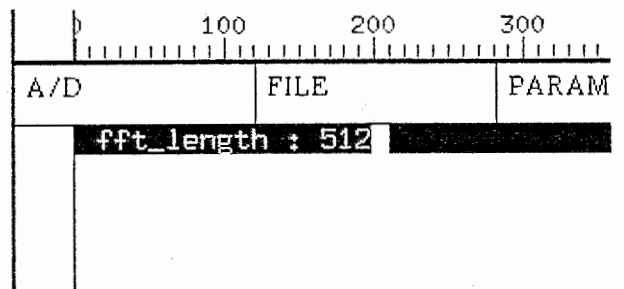
☒2.5.2.3



☒2.5.3.1



☒2.5.4.1



2.6 AREA

D/A出力及び分析の対象とする範囲を指定する。

時間軸方向に対して、逆方向に範囲を指定することも可能である。(right<left)

この場合、D/A出力は反転され、また各種分析も時間を逆行して行われる。

範囲指定時に再度AREAを選択すると、図2.6.1のようなサブメニューが表示される。

2.6.1 RESET

範囲指定を解除する。

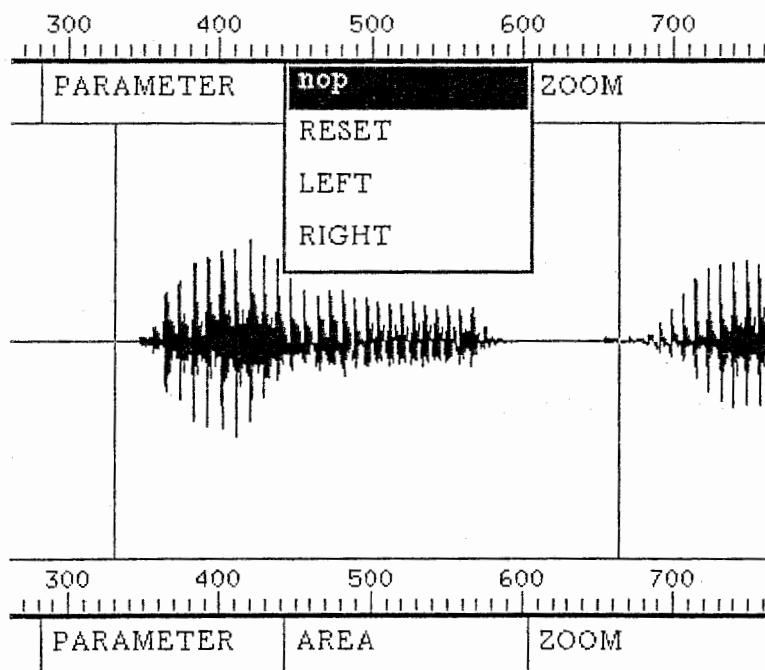
2.6.2 LEFT

左側の範囲指定を変更する。

2.6.3 RIGHT

右側の範囲指定を変更する。

図2.6.1



2.7 ZOOM

画面に表示する音声データの拡大・縮小を行う。(図2.7.1)
範囲指定時にこれを選択すると、指定されている範囲が対象となる。

2.7.1 ZOOM_UP

範囲を指定し、その区間をウィンドウ全体に拡大する。(図2.7.1.1)

2.7.2 ZOOM_DOWN

範囲を指定し、ウィンドウ全体の音声データをその区間に縮小する。

2.7.3 STANDARD

音声データの先頭から15,000ポイントをウィンドウに表示する。

図2.7.1

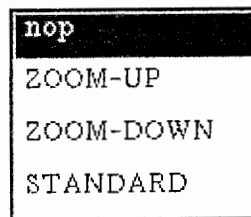
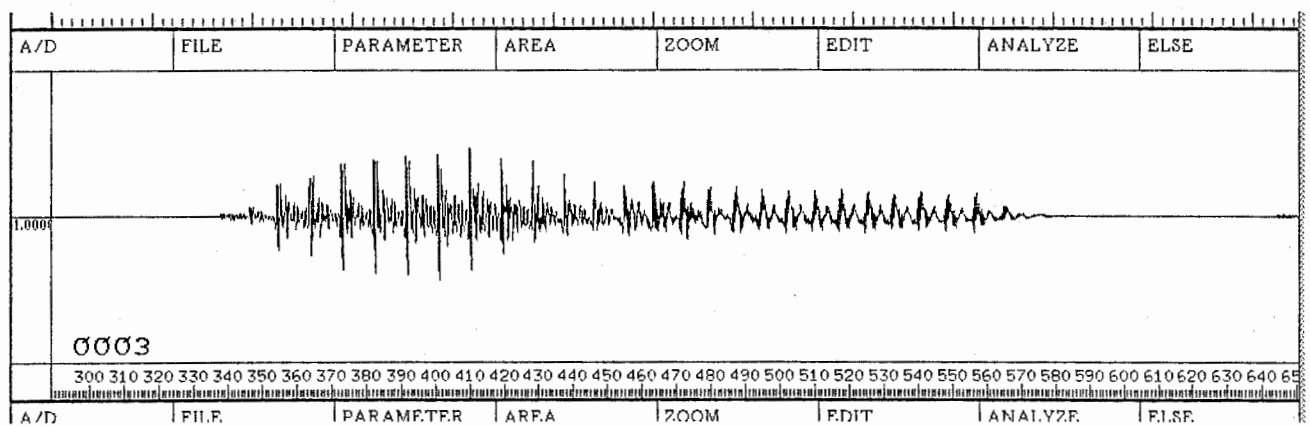


図2.7.1.1



2.8 EDIT

音声データの編集を行う。(図2.8.1)

2.8.1 COPY

範囲を指定し、その区間をカット・バッファに取り込む。取り込まれたデータは、INSERTによって取り出せる。カット・バッファは3つのウィンドウに共通であるため、ウィンドウ間でのデータの受渡しが可能である。また、範囲指定は解除される。

2.8.2 REMOVE

範囲を指定し、その区間をカット・バッファに取り込む。指定された範囲の音声データは消去される。また、範囲指定は解除される。

2.8.3 INSERT

COPYまたはREMOVEによって取り込まれたデータを、任意のウィンドウの任意の位置に出力する。

2.8.4 SILENCE

範囲を指定し、その区間を無音とする。範囲指定は解除される。

2.8.5 MIX

範囲を指定し、その区間にカット・バッファのデータを重ねる。
従ってこの操作に先立ち、copy又は、removeによって、カット・バッファにデータを取り込んでおかなければならない。
もしここで指定した範囲より、カット・バッファの方が小さい場合は、カット・バッファの大きさだけ重ね合わせることになる。

2.8.6 GAIN

音声データの振幅を変更する。
従ってこの操作に先立ち、ゲイン・インジケータを変更しておかなければならない。
範囲指定時にこの操作を行った場合は、指定されている範囲のみが、変更の対象となる。

図2.8.1

nop
COPY
REMOVE
INSERT
SILENCE
MIX
GAIN
UNDO

2.8.7 UNDO

編集中の音声データを、最後のオペレーションの前の状態に戻す。

3. アナライザ

ANALYZEを選択することによって、各種分析を行うことができる。(図3.1)
対象となるデータは、現在表示されている部分、又は指定された範囲である。

3.1 MakeSpectrum

スペクトログラムを表示する前段階として、必要なデータ計算を行う。
計算の終了はベルで知らされる。

3.2 SPECTROGRAM

3.1によって計算終了後、これを選択することによって、スペクトログラムを表示することができる。(図3.2.1)

スペクトログラムの表示を取り消す場合は、マウスカーソルをウィンドウ内に入れて、キーボードのSHIFTボタンと、マウスのMIDDLEボタンを同時にクリックする。
また波形エディタと同様、マウスのMIDDLEボタンにより、表示位置を左右に移動させることができる。

3.3 MakeAutoCorrelation

現在ウィンドウ上に表示されている区間、又は指定された範囲の音声データに対して相関計算を行う。

これは、以下に述べる、LogPower, RunningSpectrum及びLPC_SPECTRUM_SLICEに先立って計算しておかなければならない。

計算終了は、ベルで知らされる。

3.4 LOG_POWER

3.3により計算された相関データを基に、ログパワーを計算して、画面上に表示する。(図3.4.1)

マウスのクリックにより、以下の通り表示位置及び大きさを任意に設定できる。

- ・左ボタン 音声データの指定された位置に重ねて表示する。
- ・中央ボタン 任意の位置に任意の大きさのウィンドウを作成する。
- ・右ボタン ウィンドウの位置のみ指定する。大きさは左ボタンと同じである。

ログパワーの表示を取り消す場合は、マウスカーソルをウィンドウ内に入れて、キーボードのSHIFTボタンと、マウスのMIDDLEボタンを同時にクリックする。
表示ウィンドウはresizeすることができる。

3.5 Running_Spectrum

3.3により計算された相関データを基に、ランニング・スペクトルを計算して、画面上に表示する。(図3.5.1)

3.4と同様に、マウスのクリックにより、表示位置及び大きさを任意に設定できる。ランニング・スペクトルの表示を取り消す場合は、マウスカーソルをウィンドウ内に入れて、キーボードのSHIFTボタンと、マウスのMIDDLEボタンを同時にクリックする。

表示ウィンドウはresizeすることができる。

3.6 FFT_SPECTRUM_SLICE

マウスにより、分析の対象となる位置を指定する。(図3.6.1)

マウスのクリックにより、以下の通り表示位置及び大きさを任意に設定できる。

- ・左ボタン 音声波形ウィンドウの左端に150x150の大きさで表示する。
- ・中央ボタン 任意の位置に任意の大きさのウィンドウを作成する。
- ・右ボタン ウィンドウの位置のみ指定できる。大きさは150x150である。

表示ウィンドウはresizeすることができる。

指定した位置表示は、4.2REDRAWにより消去できる。

FFTスペクトラム・スライスの表示を取り消す場合は、マウスカーソルをウィンドウ内に入れて、キーボードのSHIFTボタンと、マウスのMIDDLEボタンを同時にクリックする。

3.7 LPC_SPECTRUM_SLICE

3.3により計算された相関データを基に、LPC・スペクトラム・スライスを計算して、画面上に表示する。(図3.7.1) 故に、下記の分析の位置指定は、相関計算した範囲にかぎる。(範囲外の指定はベルで知らされる。)

マウスにより、分析の対象となる位置を指定する。

マウスのクリックにより、以下の通り表示位置及び大きさを任意に設定できる。

- ・左ボタン 音声波形ウィンドウの左端に150x150の大きさで表示する。
- ・中央ボタン 任意の位置に任意の大きさのウィンドウを作成する。
- ・右ボタン 任意のウィンドウの位置のみ指定する。大きさは150x150である。

表示ウィンドウはresizeすることができる。

指定した位置表示は、4.2REDRAWにより消去できる。

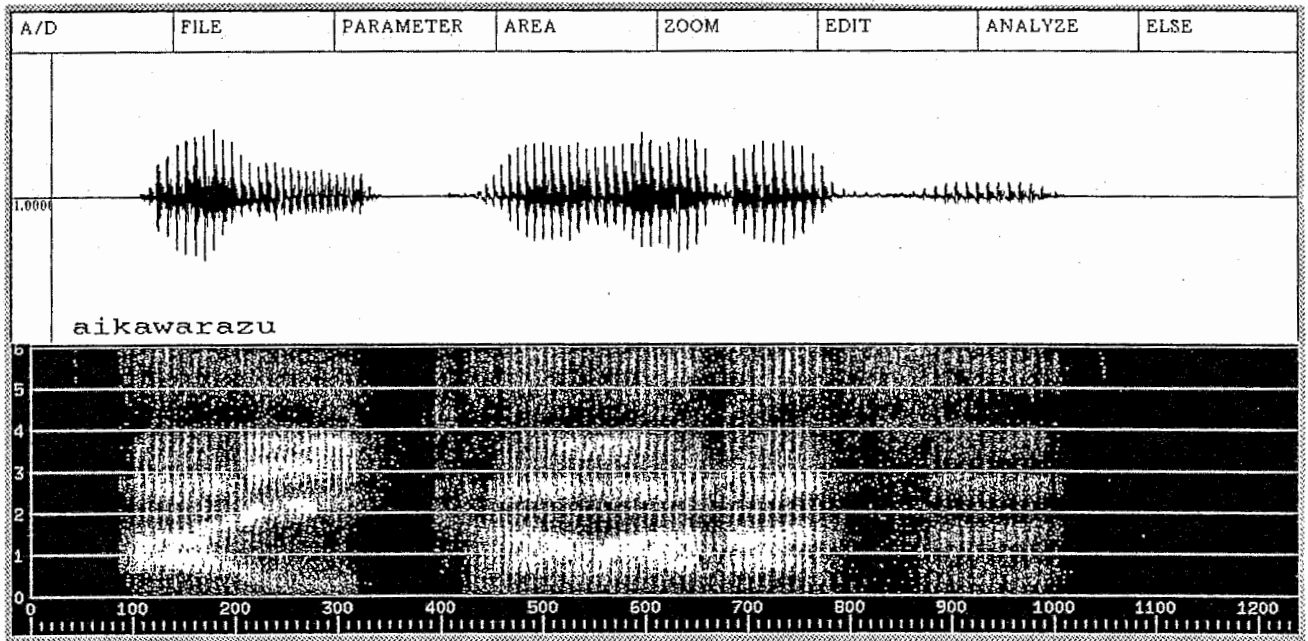
LPCスペクトラム・スライスの表示を取り消す場合は、マウスカーソルをウィンドウ内に入れて、キーボードのSHIFTボタンと、マウスのMIDDLEボタンを同時にクリックする。

☒3.1

```

nop
MakeSpectrum
SPECTROGRAM
MakeAutoCorrela
LOG<POWER
RUNNING<SPEC
FFT<SPECTRUM
LPC<SPECTRUM
    
```

☒3.2.1



☒3.4.1

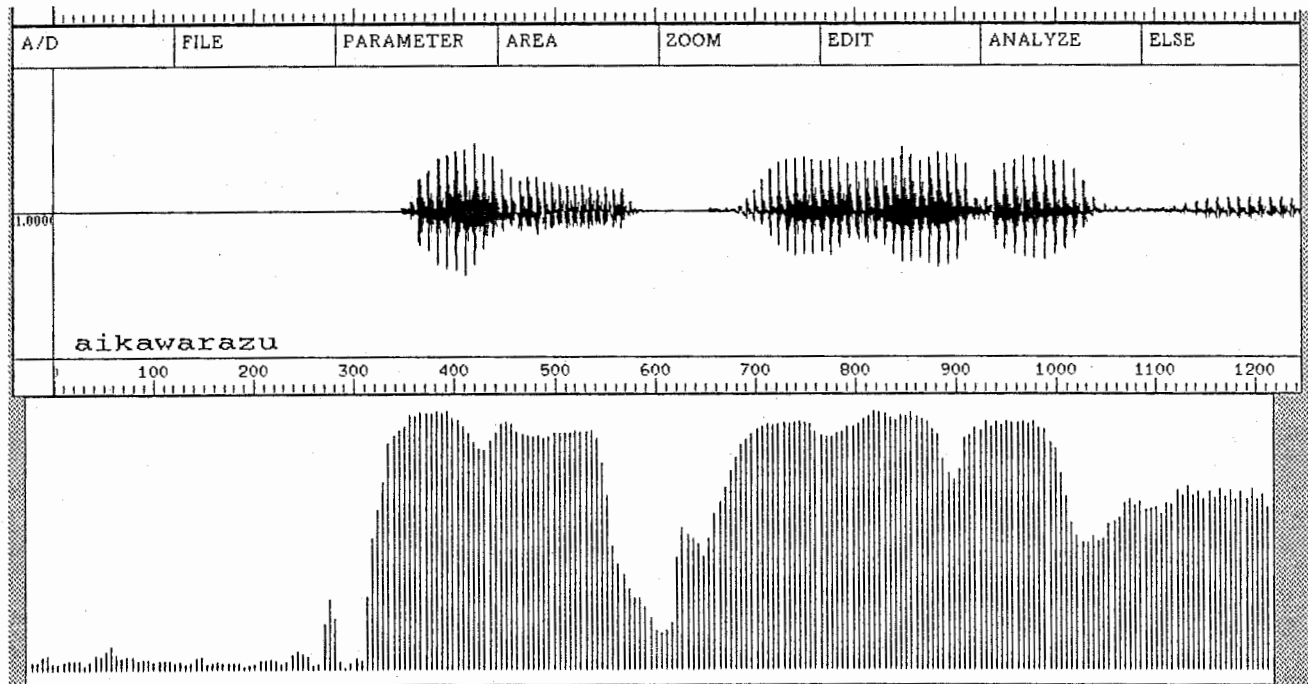


図3.5.1

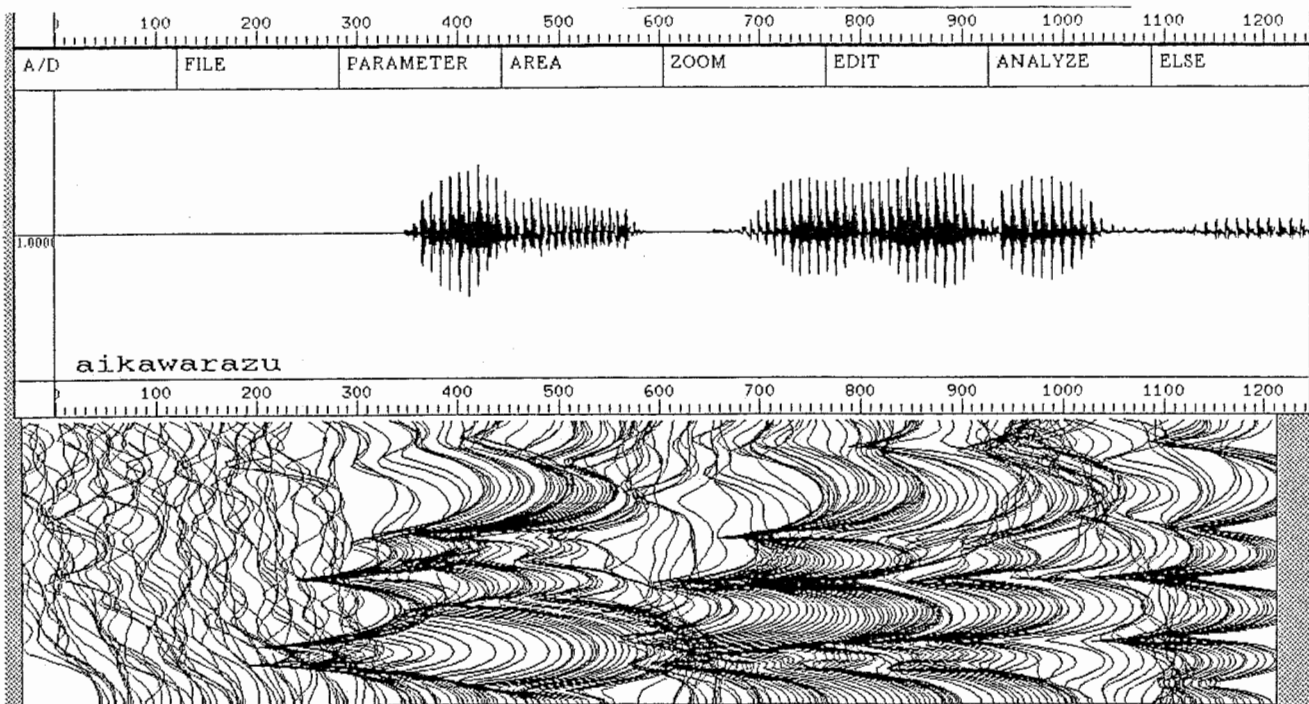


図3.6.1

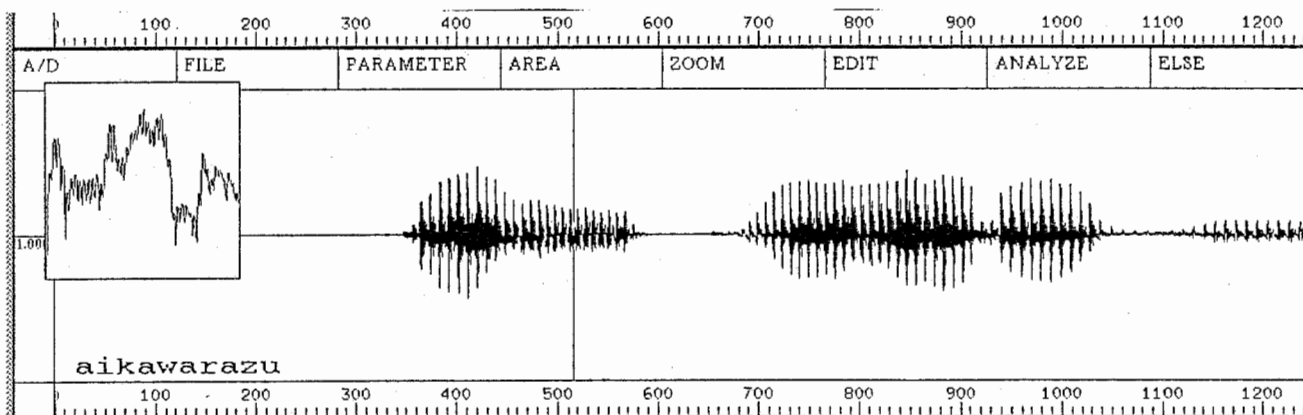
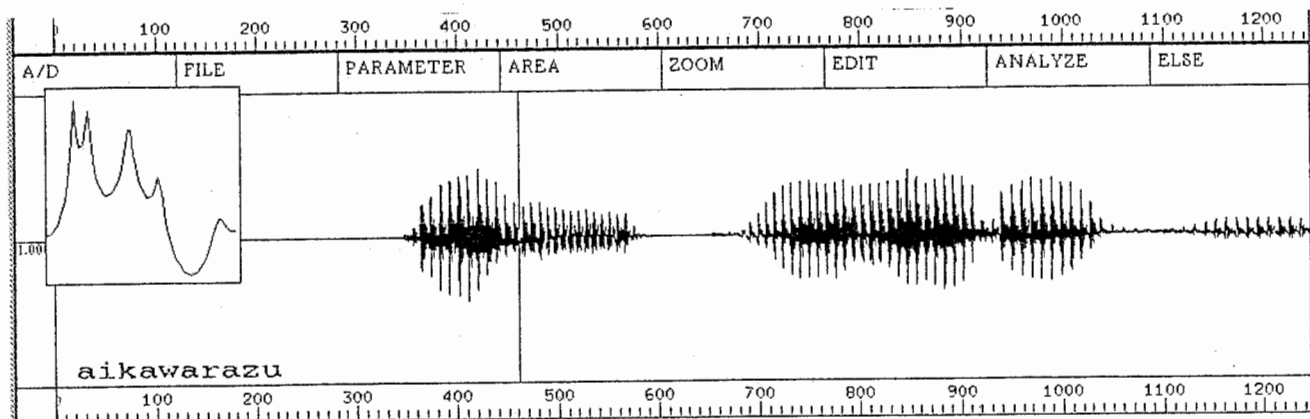


図3.7.1



4. その他の機能

ELSEを選択することによって、便利な機能を使用できる。(図4.1)

4.1 LABEL

2.4.4でラベルファイル名を指定してある場合に、これを選択すると、音声データに重ねて、ラベルが表示される。(図4.1.1)

ラベルを表示してある状態で再度選択すると、表示解除となる。

4.2 REDRAW

何らかの原因により、ウィンドウが破壊された場合又は、パラメータを変更した場合に、音声波形を再表示させるものである。

4.3 GENERATER

周波数を指定し、正弦波を発生させるものである。(図4.3.1) (図4.3.2)

サンプリング周波数は、2.51. sampling_rateによって入力された値を使用する。

4.4 LOCATION

現在画面上に出力されている部分を、パワー包絡により示す。(図4.4.1)
再度クリックする事により表示を取り消すことができる。

4.5 ICONS

画面上に散乱しているアイコンを、整列させるものである。

4.6 CLEAR

バッファ上のデータをクリアする。

4.7 WINDOW

波形編集用のウィンドウを追加する。最大3画面である。

図4.1

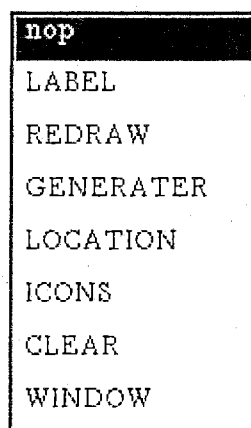


図4.1.1

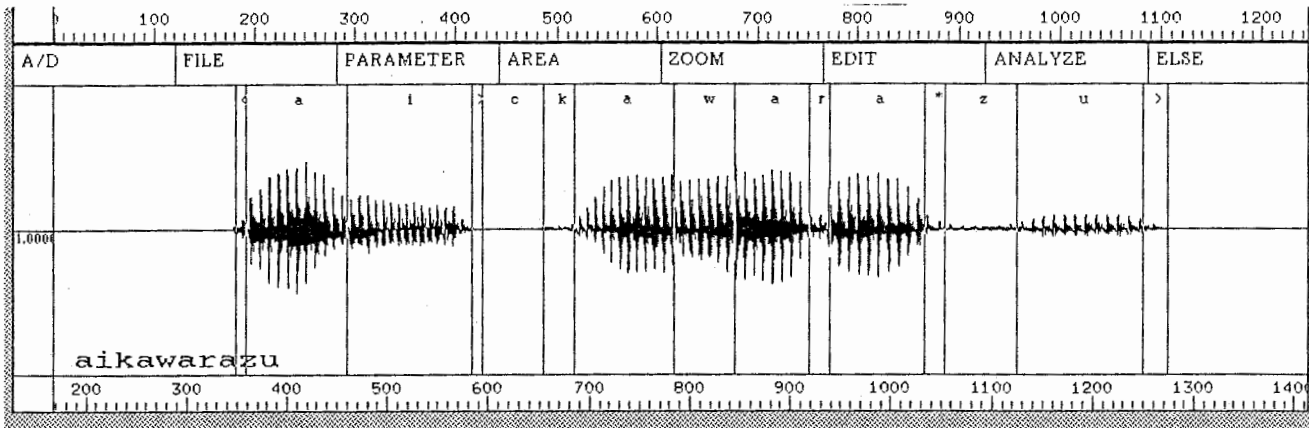


図4.3.1

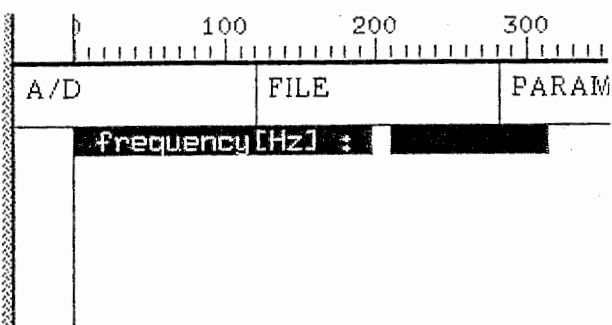


図4.3.2

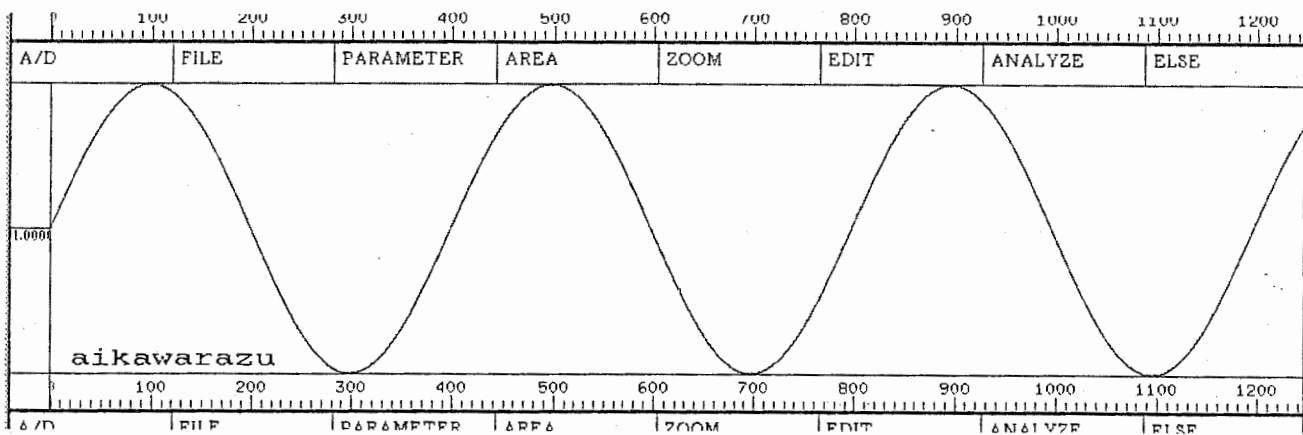
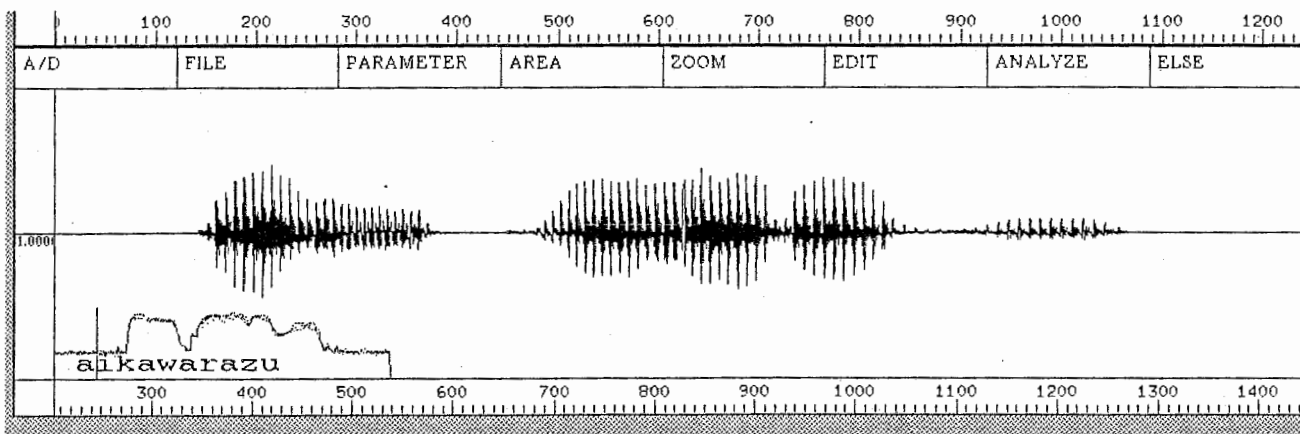


図4.4.1



Appendix-A セットアップの仕方

ワークベンチVer3.0では、D/A出力や、アナライズの為のプログラムが独立している。従ってそれら全ての実行ファイルを /usr/local/bin ディレクトリに移動しておく必要がある。

```
% mv wedt /usr/local/bin
% mv daout /usr/local/bin
% mv spect_cul /usr/local/bin
% mv spect_out /usr/local/bin
% mv xcleanup /usr/local/bin
% mv AutoCorrel /usr/local/bin
% mv RunSpect /usr/local/bin
% mv LogPower /usr/local/bin
% mv LPCslice /usr/local/bin
% mv FETslice /usr/local/bin
```

もし /usr/local/bin 以外のディレクトリに置きたい場合は、.Xdefaults ファイルに以下の例に習って、そのディレクトリを指定しなければならない。

(例) /usr/users/bin の下に置いた場合。
wedt.bin: /usr/users/bin

Appendix-B ラインエディット機能について

save ファイル名、ラベルファイル名及び各種パラメータの入力時に、以下のラインエディット機能が使用できる。

```
cntrl_A 文字列の先頭へカーソルを移動する。
cntrl_B 一文字左へカーソルを移動する。
cntrl_D カーソルのある一文字を消去する。
cntrl_E 文字列の最後にカーソルを移動する。
cntrl_F 一文字右へカーソルを移動する。
cntrl_K カーソルの位置より右側を消去する。
```


第二部 プログラム仕様書

目 次

1. 概要		
1.1 機能	1
1.2 処理の流れ	1
2. ソフトウェア構成		
2.1 概要	2
2.2 ブロック・チャート		
1. 波形エディット・プロセス	3
2. 相関計算(AutoCorrel)	20
3. ログパワー(LogPower)	20
4. ランニング・スペクトログラム(RunSpect)	20
5. FFTスライス・スペクトログラム(FFTslice)	21
6. LPCスライス・スペクトログラム(LPCslice)	21
2.3 各関数の説明		
1. 波形エディタの関数		
(1) ADIN (ad_in.c)	22
(2) ad_in (ad_in.c)	22
(3) amplify (X_io.c)	22
(4) analyze (analyze.c)	23
(5) append_string (file_io.c)	23
(6) area (area.c)	23
(7) CloseADIN (ad_in.c)	24

(8) Copy_WaveForm (X_io.c)	24
(9) CreateWindows (windows.c)	24
(10) conv_p (file_io.c)	25
(11) cul_power (cul_power.c)	25
(12) cursor_set (cursor.c)	25
(13) cut (edit.c)	26
(14) DestroyXform (X_io.c)	26
(15) DrawPow (draw_power.c)	27
(16) da_out (da_out.c)	28
(17) deiconify (X_io.c)	28
(18) direx (direx.c)	28
(19) disp_logpower (analyze.c)	29
(20) disp_runspect (analyze.c)	29
(21) disp_spectrum (analyze.c)	29
(22) draw (X_io.c)	30
(23) draw_bar (X_io.c)	30
(24) draw_filename (X_io.c)	31
(25) draw_horizon_line (X_io.c)	31
(26) draw_icon (X_io.c)	31
(27) draw_label (X_io.c)	32
(28) draw_power (draw_power.c)	32
(29) draw_scale (X_io.c)	33
(30) draw_wave (X_io.c)	33

(31)	edit (edit.c)	34
(32)	FreeLabel (FreeLabel.c)	34
(33)	file (file.c)	34
(34)	flag (flag.c)	35
(35)	func_sw (X_io.c)	35
(36)	gain (edit.c)	36
(37)	get_default (X_io.c)	36
(38)	get_event (X_io.c)	37
(39)	get_exec (X_io.c)	37
(40)	h_bar (X_io.c)	38
(41)	h_line (X_io.c)	38
(42)	Init_CreateWindows (windows.c)	39
(43)	iconify (X_io.c)	39
(44)	init_AnIPID (file_io.c)	39
(45)	init_waveform (file_io.c)	40
(46)	init_window (X_io.c)	40
(47)	insert (edit.c)	40
(48)	KillChild (X_io.c)	41
(49)	label (label.c)	41
(50)	label_name (file.c)	41
(51)	load (file.c)	42
(52)	load_file (file.c)	42
(53)	locate_power (locate_power.c)	43

(54)	MakeMenu (direx.c)	43
(55)	MapSubwindows (windows.c)	44
(56)	main (main.c)	44
(57)	make_correl (analyze.c)	44
(58)	make_flag (flag.c)	45
(59)	make_spectrum (analyze.c)	45
(60)	memory_clr (file_io.c)	45
(61)	memory_cpy (file_io.c)	46
(62)	memory_mix (file_io.c)	47
(63)	memory_mul (file_io.c)	48
(64)	mix (edit.c)	48
(65)	OpenADIN (ad_in.c)	49
(66)	PutScale (PutScale.c)	50
(67)	param (param.c)	51
(68)	path (file.c)	51
(69)	pickup (pickup.c)	52
(70)	power (power.c)	53
(71)	ResizeParentWindow (windows.c)	53
(72)	redraw (X_io.c)	54
(73)	redraw_func (X_io.c)	54
(74)	redraw_wave (X_io.c)	54
(75)	reset_amp (X_io.c)	55
(76)	reset_form (X_io.c)	55

(77)	reset_wdata (X_io.c)	55
(78)	SelectInput (X_io.c)	56
(79)	save (file.c)	56
(80)	save_file (file.c)	57
(81)	set_fft_slice (analyze.c)	57
(82)	set_lpc_slice (analyze.c)	58
(83)	set_path (file_io.c)	58
(84)	set_path_name (file_io.c)	59
(85)	shift (file_io.c)	59
(86)	silence (edit.c)	60
(87)	sin_gen (sin.c)	61
(88)	sonota (sonota.c)	61
(89)	sure (X_io.c)	62
(90)	time_to_point (file_io.c)	62
(91)	v_line (X_io.c)	63
(92)	work_file (file_io.c)	63
(93)	X_fg (draw_power.c)	64
(94)	zero_count (file_io.c)	65
(95)	zero_cross (file_io.c)	65
(96)	zombie (analyze.c)	65
(97)	zoom (zoom.c)	66
(98)	zoom_down (zoom.c)	66
(99)	zoom_std (zoom.c)	66

(100) zoom_up (zoom.c)	67
------------------------	-------	----

2. アナライザの関数

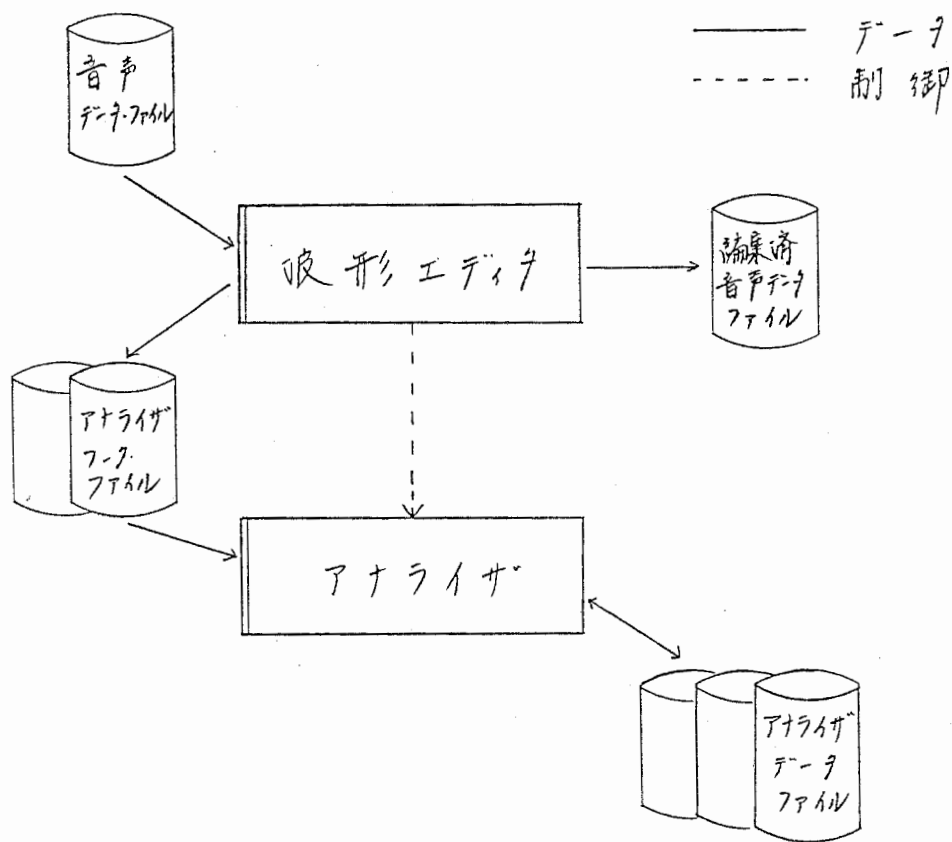
(1) アナライザ共通関数	68
(2) 相関計算 (AutoCorrel.c)	69
(3) ログ・パワー出力 (Logpower.c)	69
(4) ランニング・スペクトログラム出力 (RunSpect.c)	70
(5) FFTスライス・スペクトログラム出力 (FFTslice.c)	70
(6) LPCスライス・スペクトログラム出力 (LPCslice.c)	71

1. 概要

1.1 機能

本ソフトウェア(以後WBと略す)は、音声データの編集・加工を行う。
又、各アナライザ(分析・解析)の起動管理を行う。

1.2 処理の流れ



アナライザは、子プロセスとして起動される。

2. ソフトウェア構成

2.1 概要

WBは、波形エディタとアナライザ(スペクトログラム演算/出力、相関計算、ログ・パワー、ランニング・スペクトログラム、LPC/FFTスライス出力)プロセスの計8プロセスで構成されている。

尚、スペクトログラム演算/出力の2プロセスに関しては、別紙参照こと。

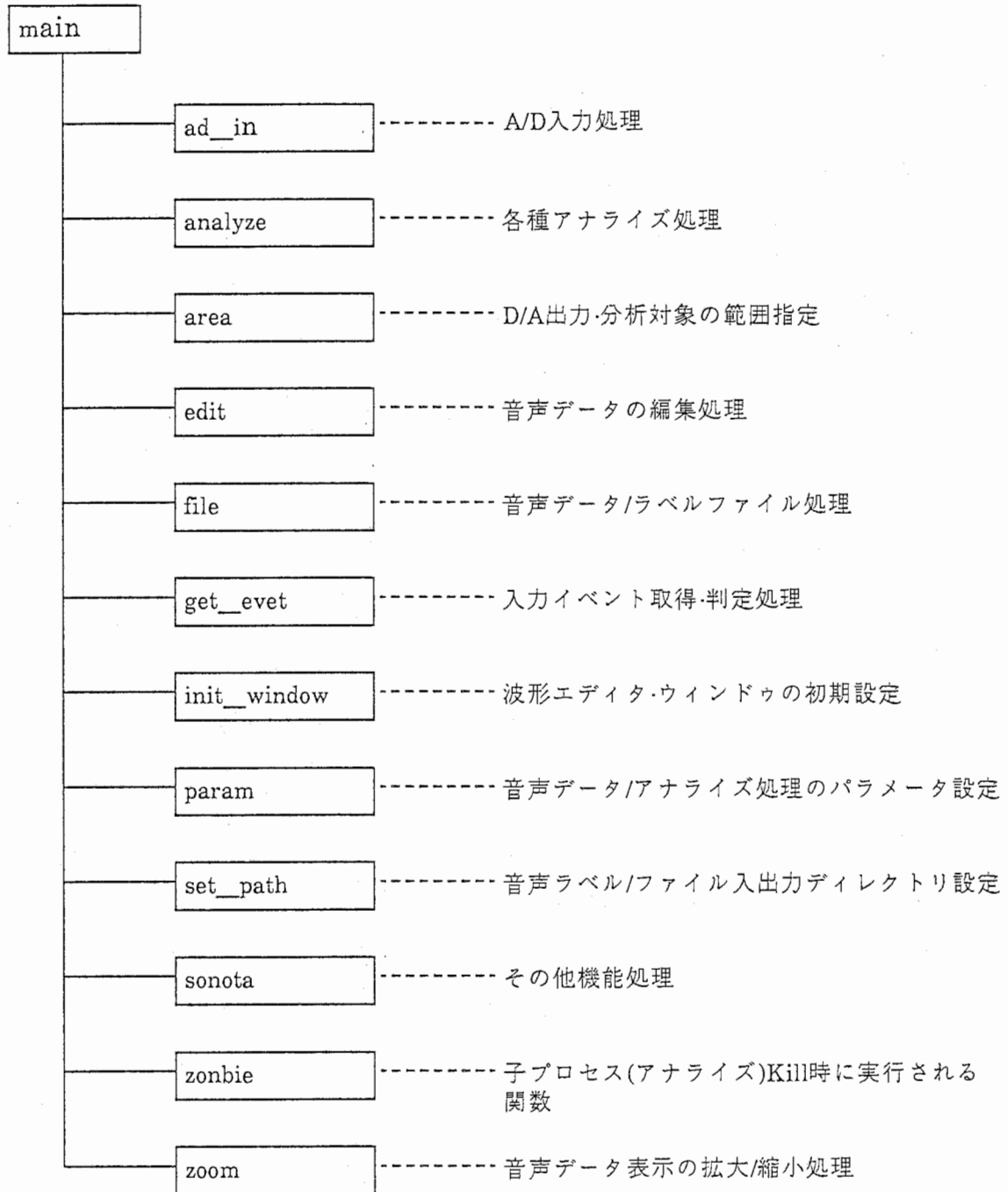
X、LPC、Xカーネル/ウィンドウ・ユーティリティの各ライブラリーをコールしている。

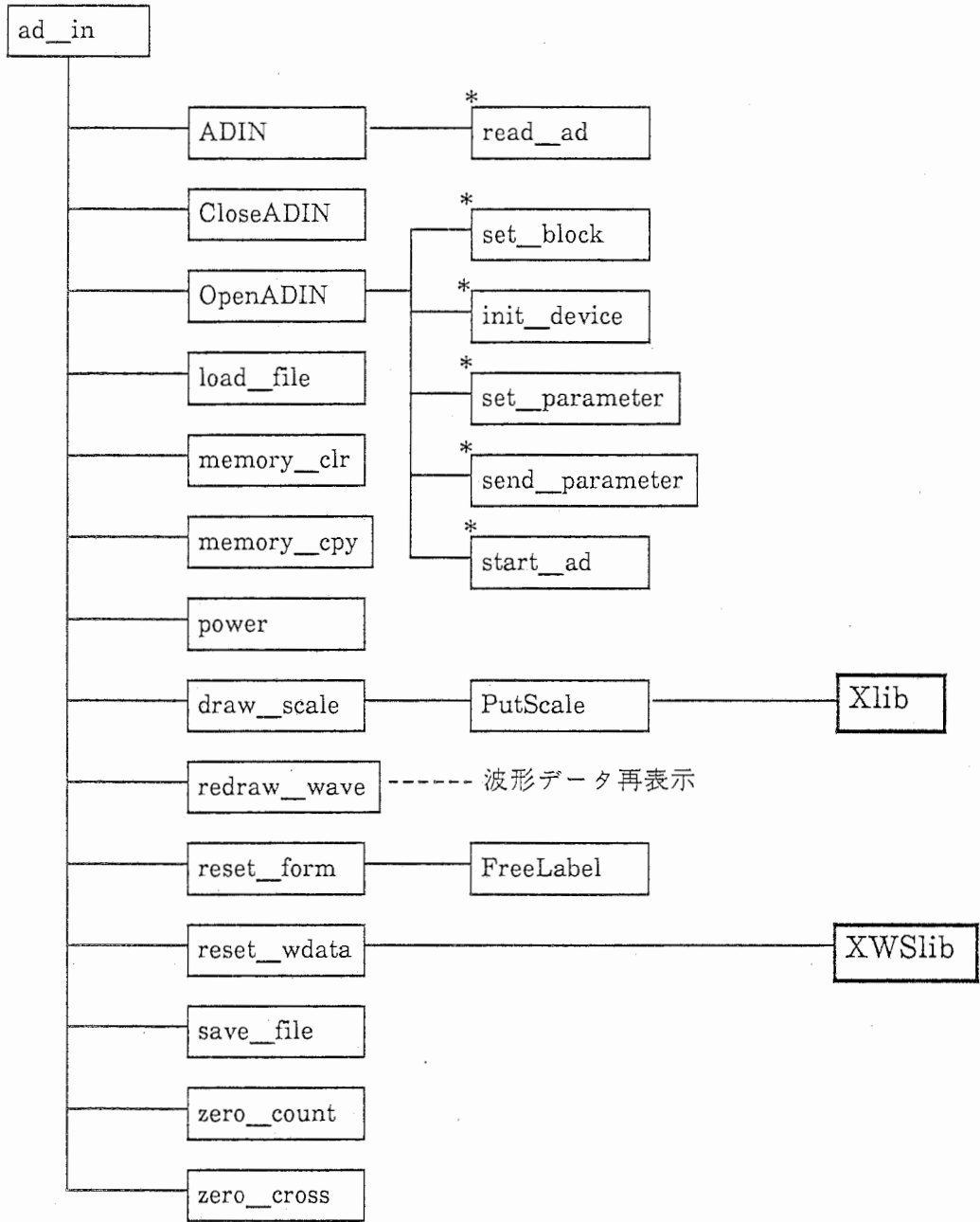
2.2 ブロック・チャート

プロセス別に、モジュールの関連を次頁以下に記す。

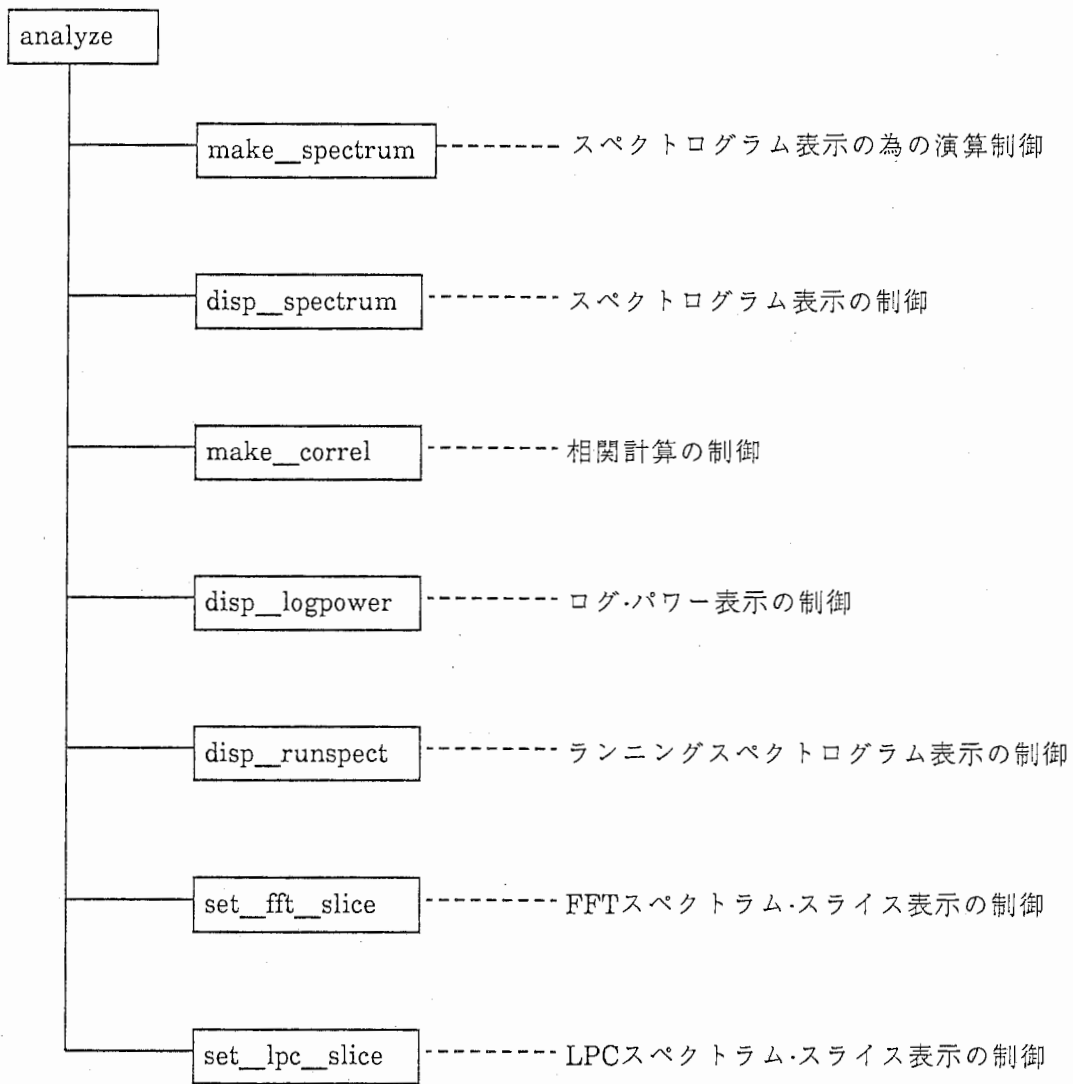
1. 波形エディット・プロセス

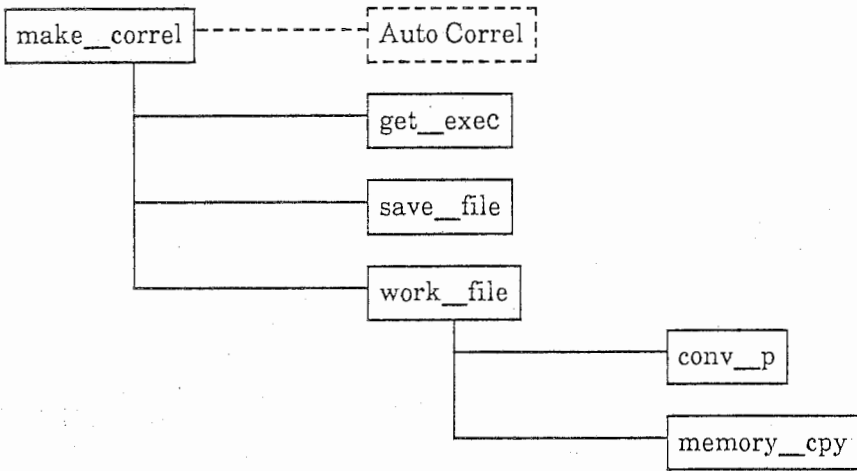
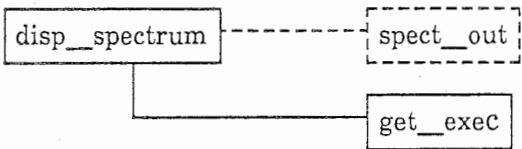
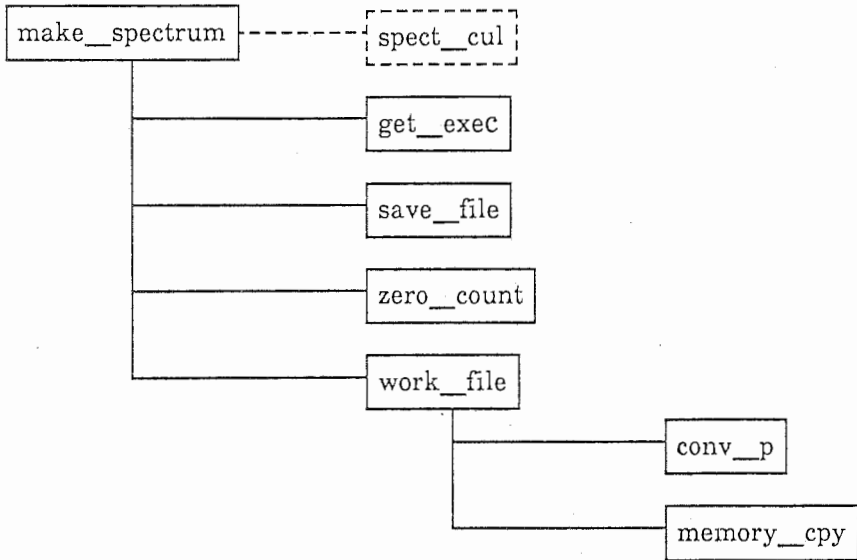
 は、子プロセスとして起動させるものである。

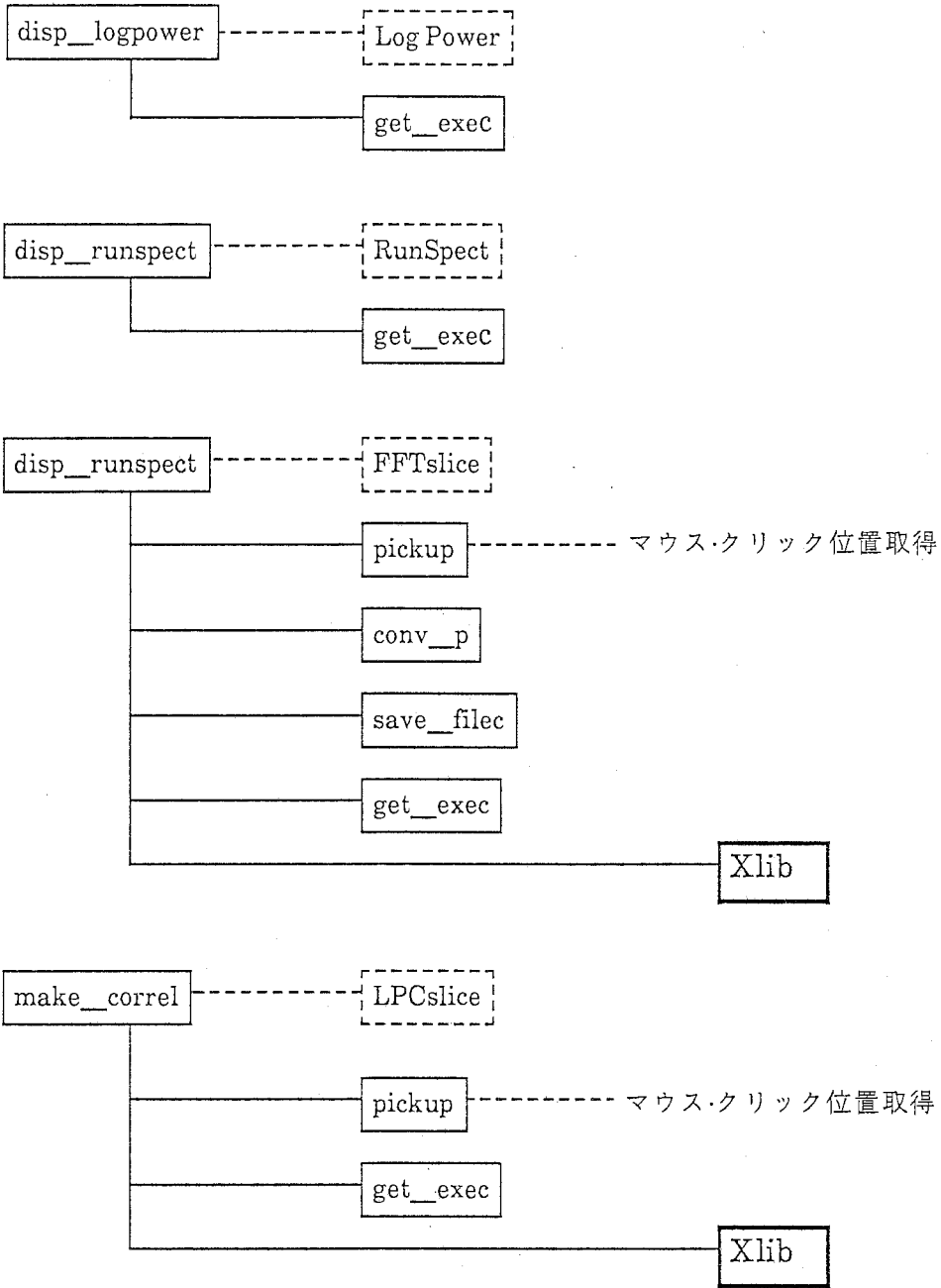


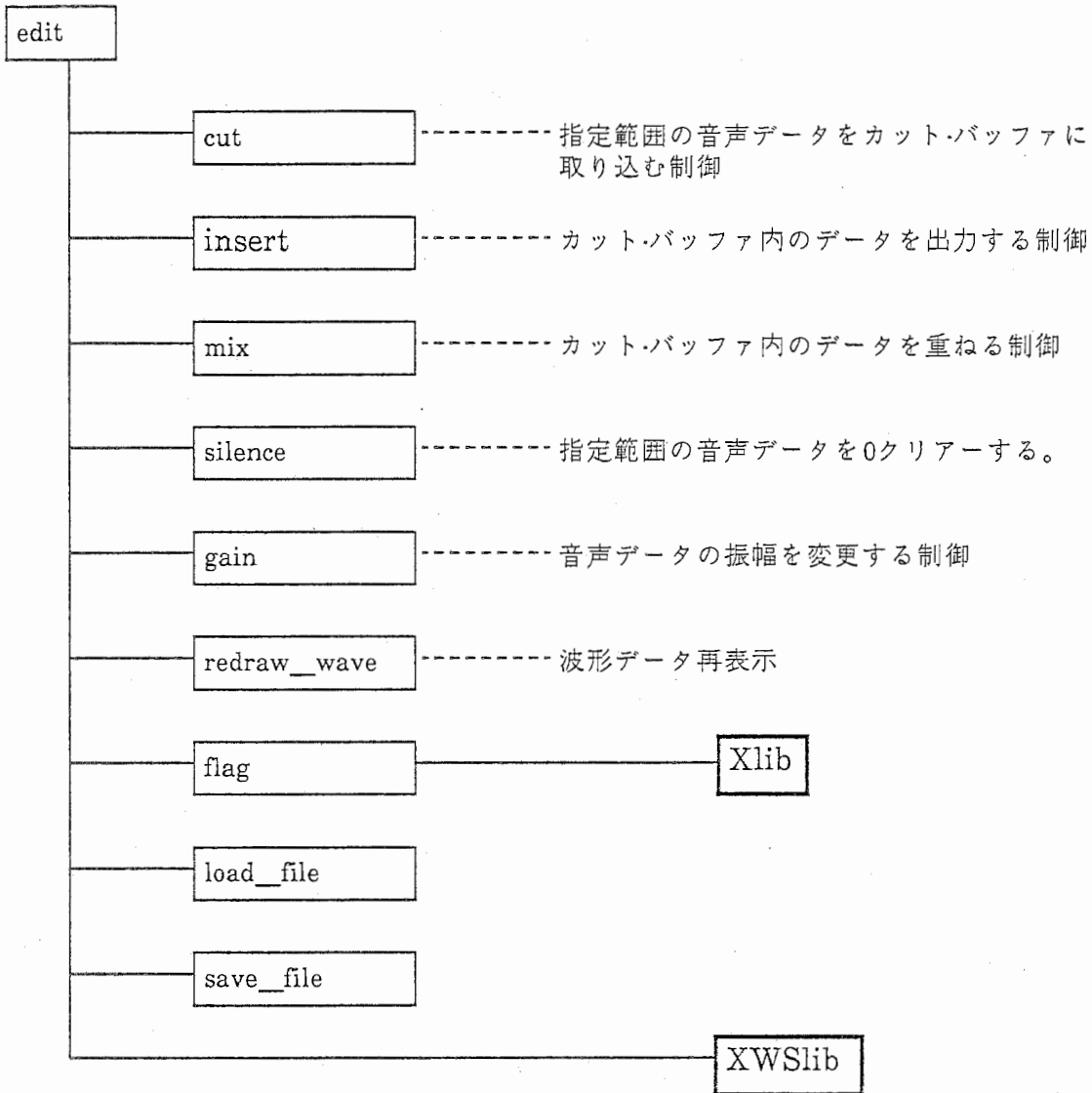
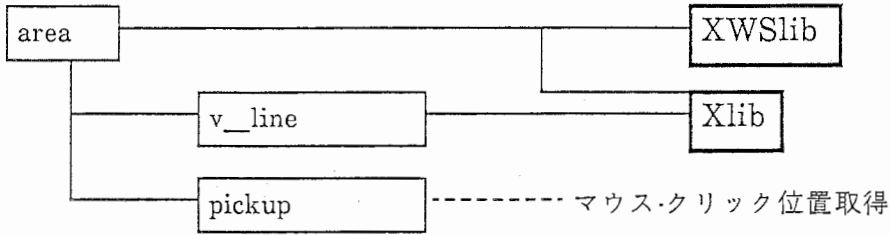


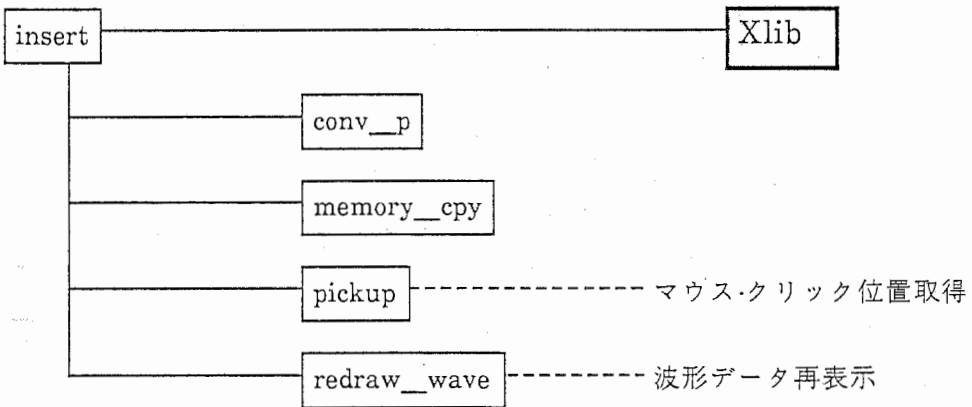
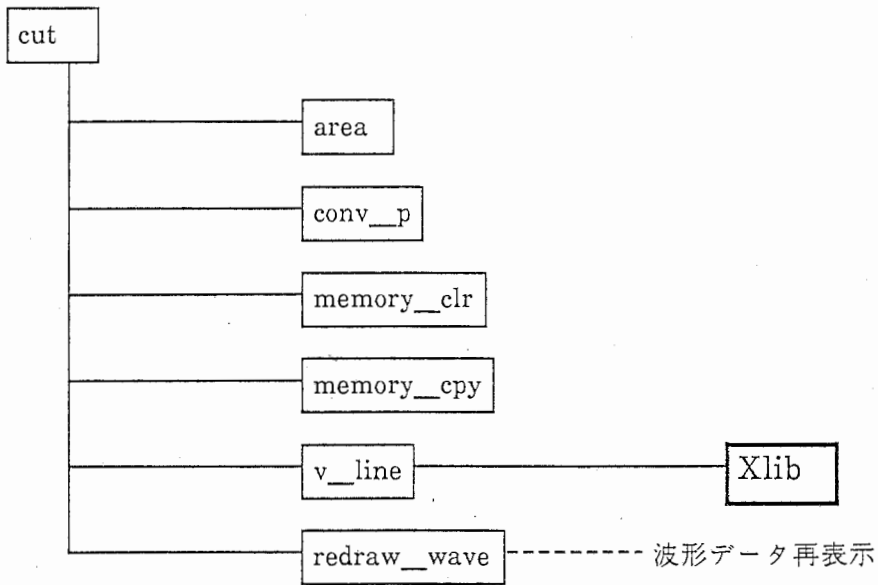
*印付きのモジュールは、 μ DASBOX基本サブルーチンを示す。
 (本マニュアルでの説明は除く。詳細は、 μ DASBOX基本サブルーチン(ULTRIX-32版)使用
 手引書参照の事。)

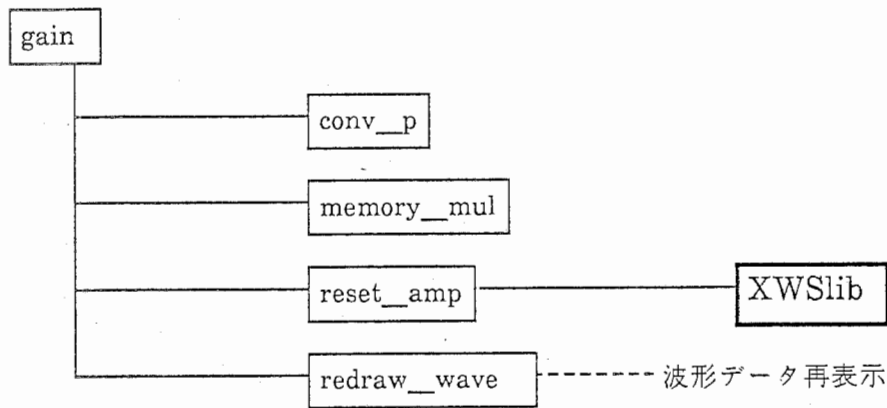
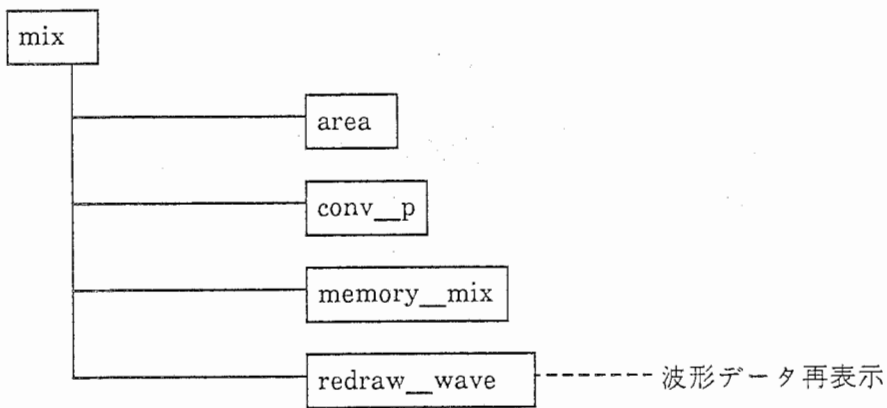
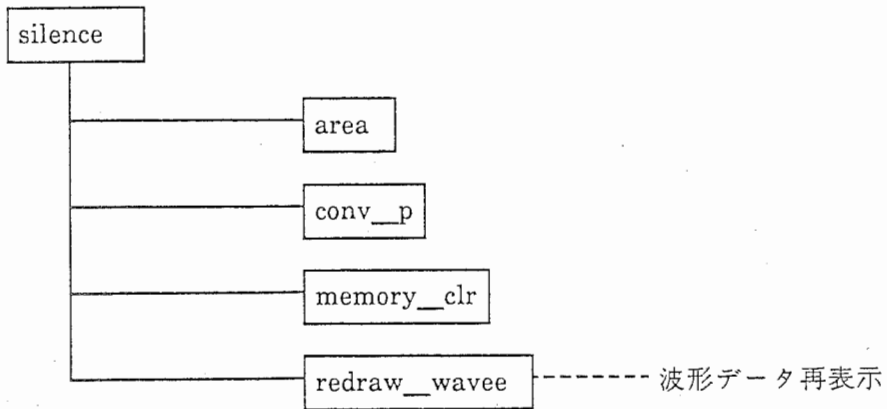


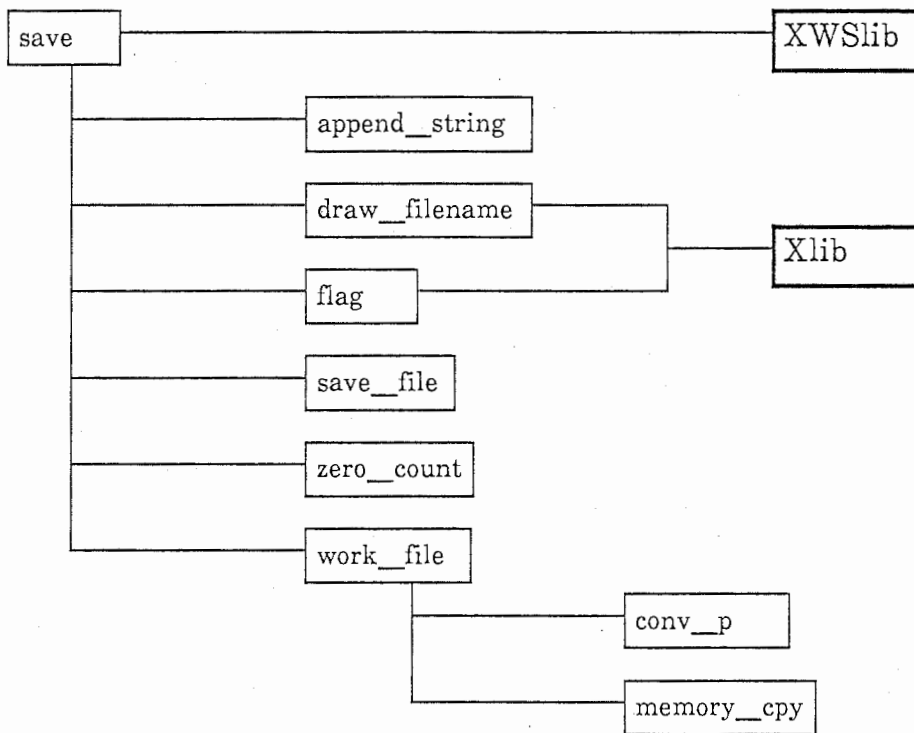
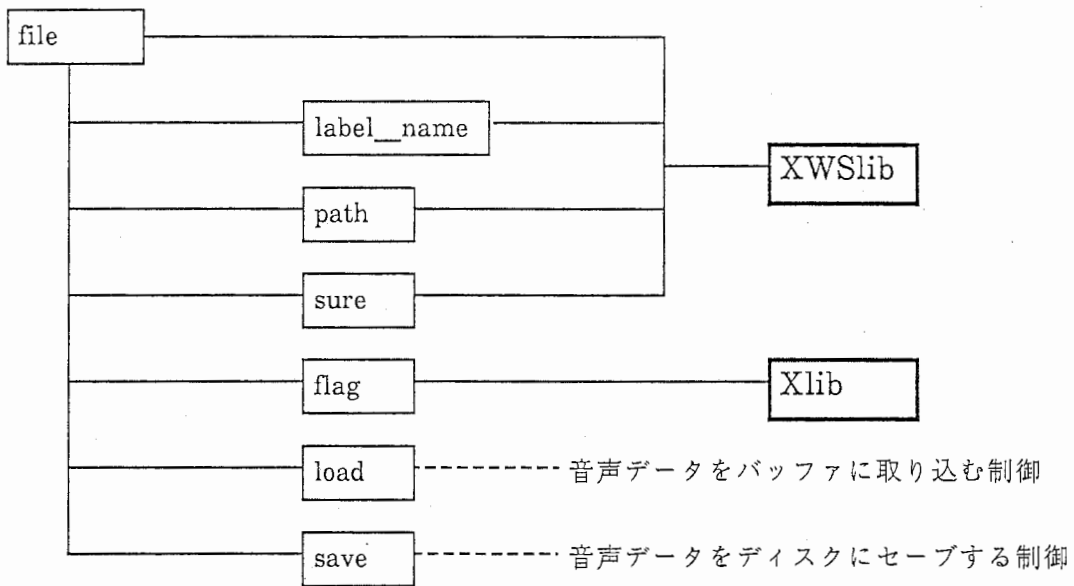


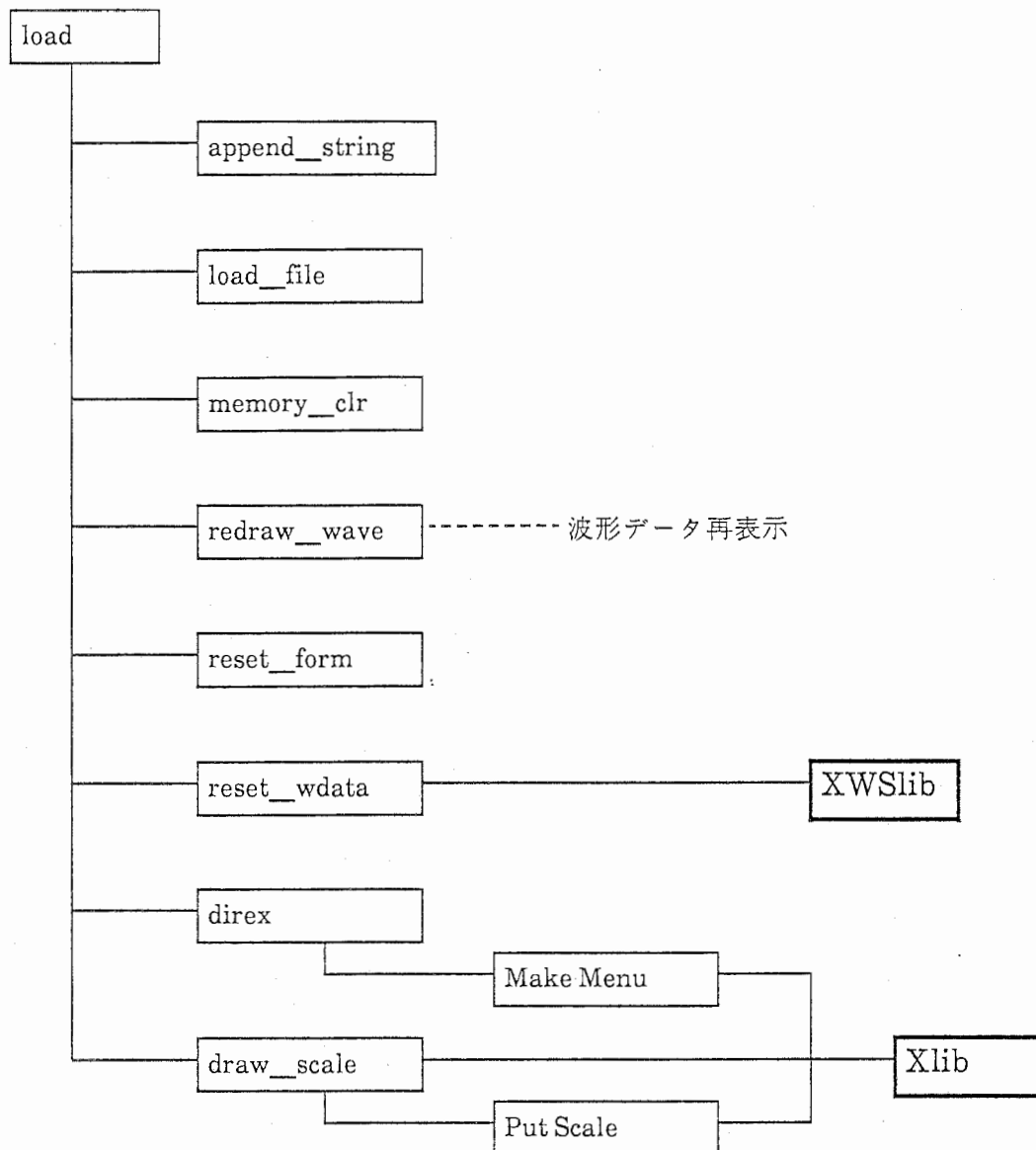


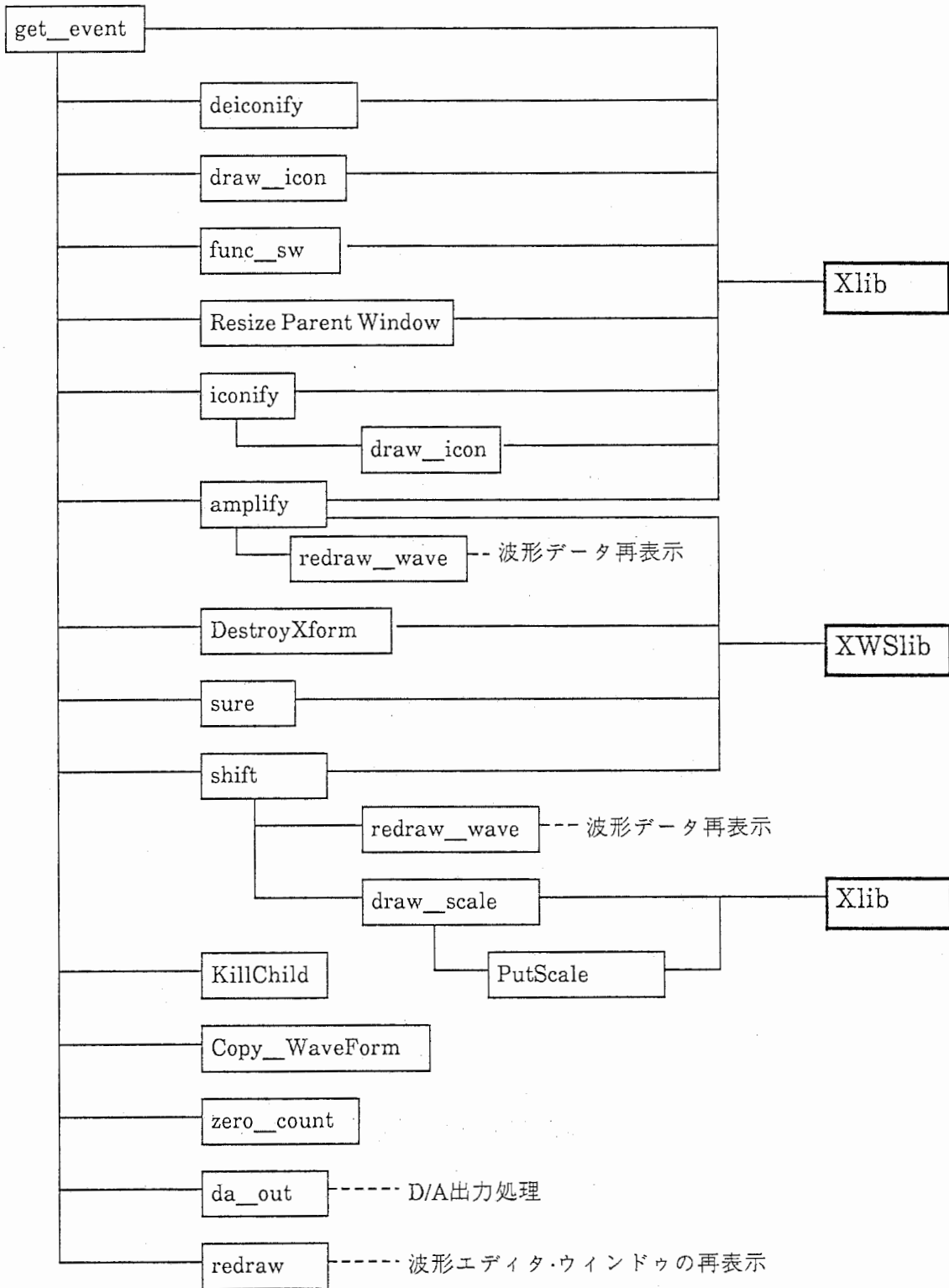


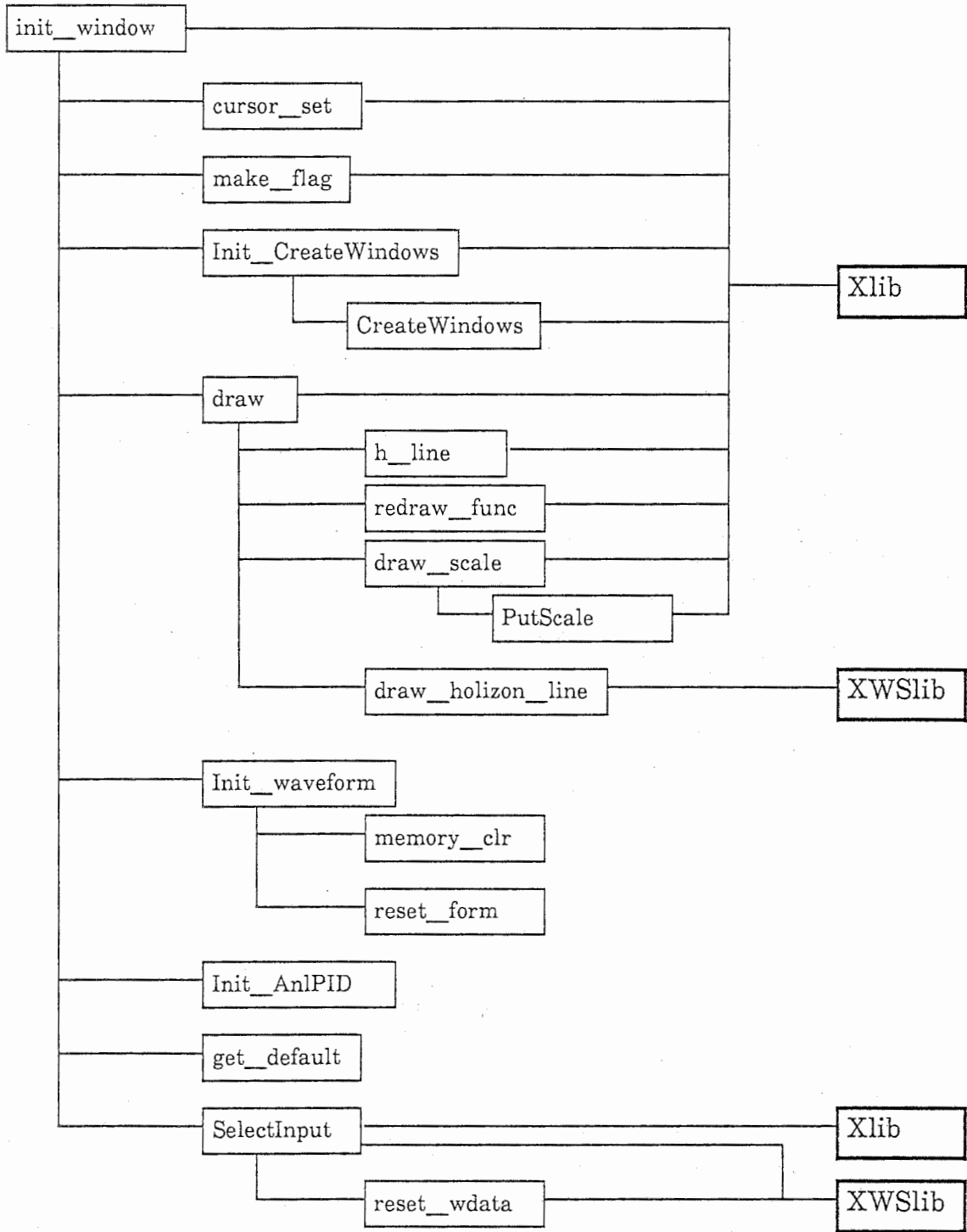


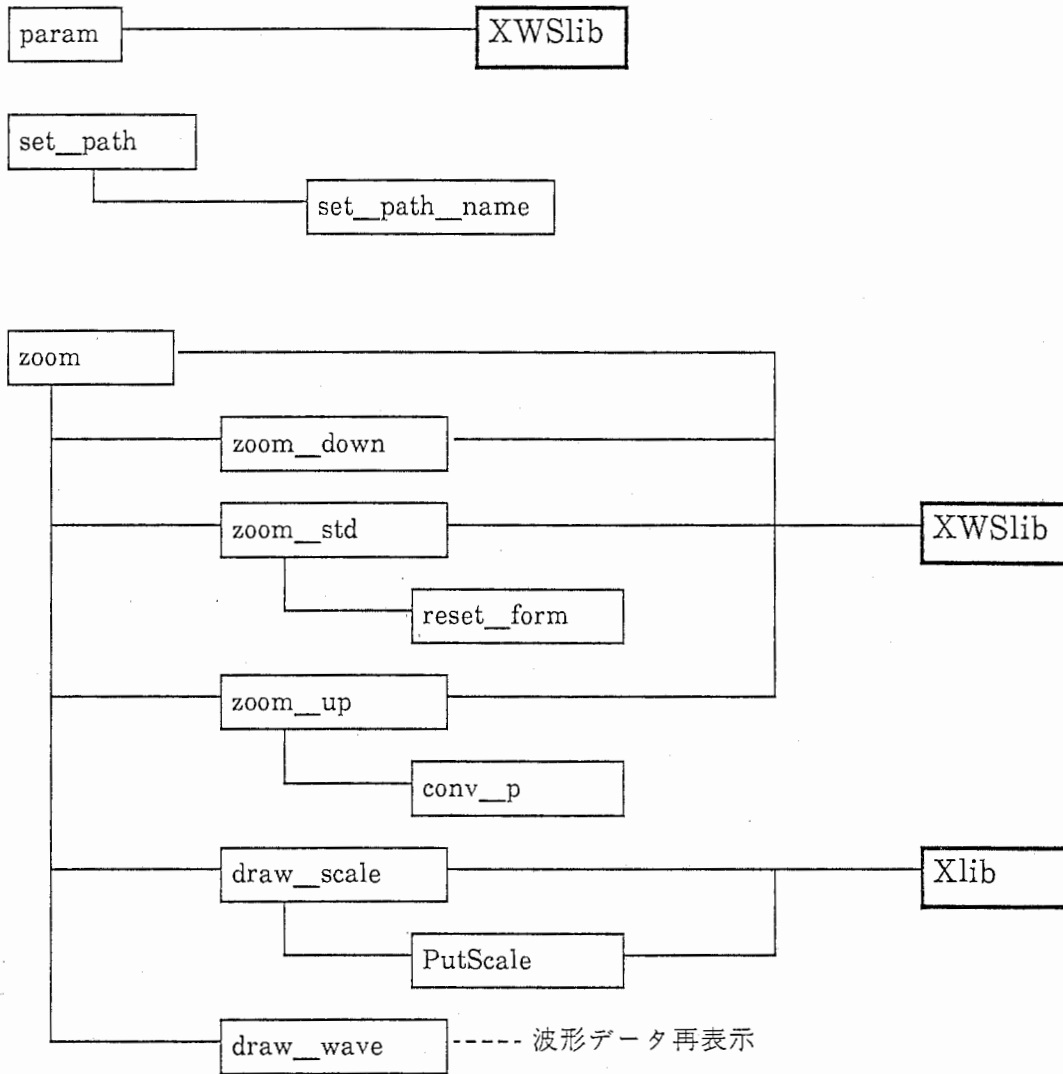


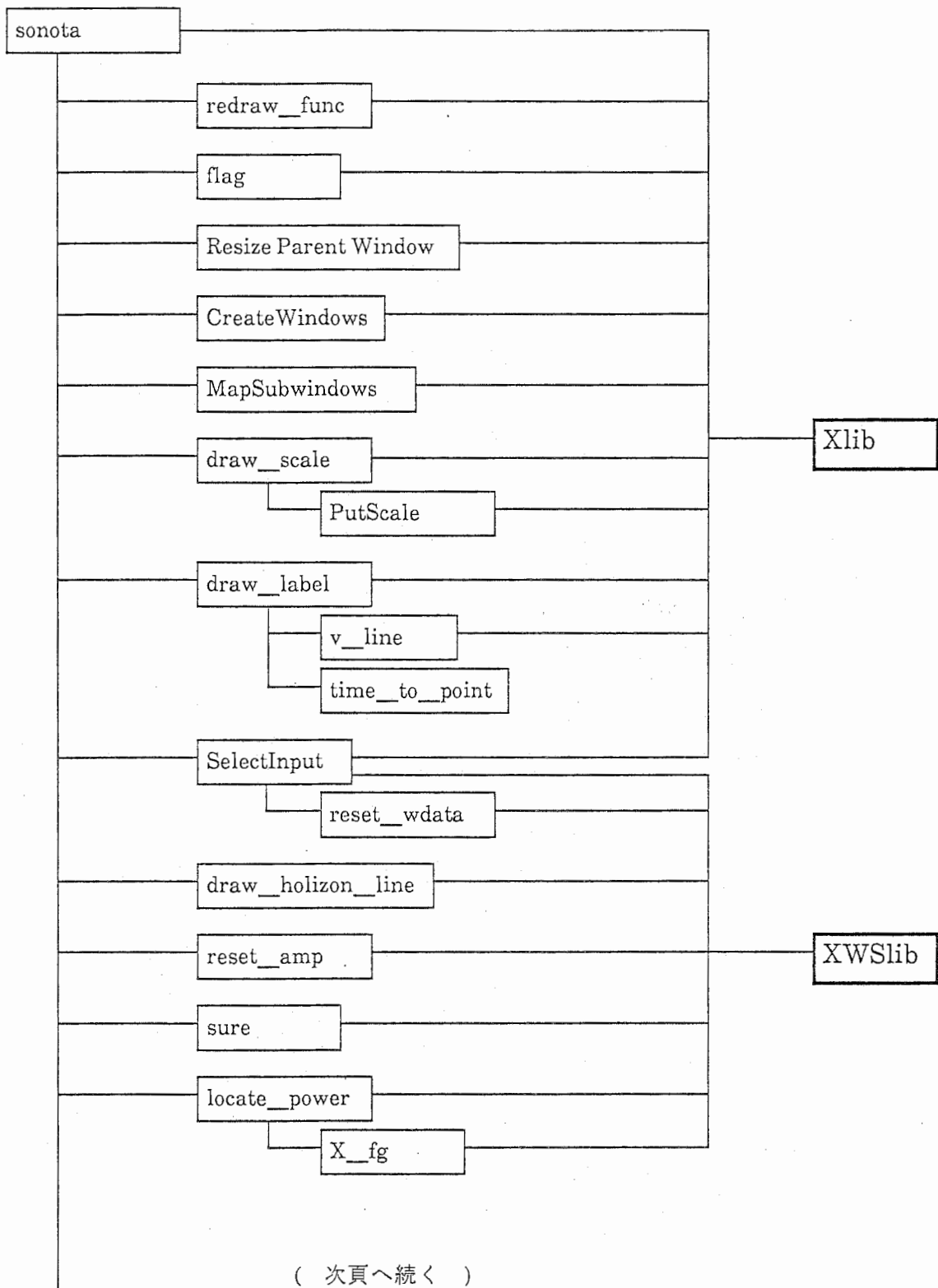


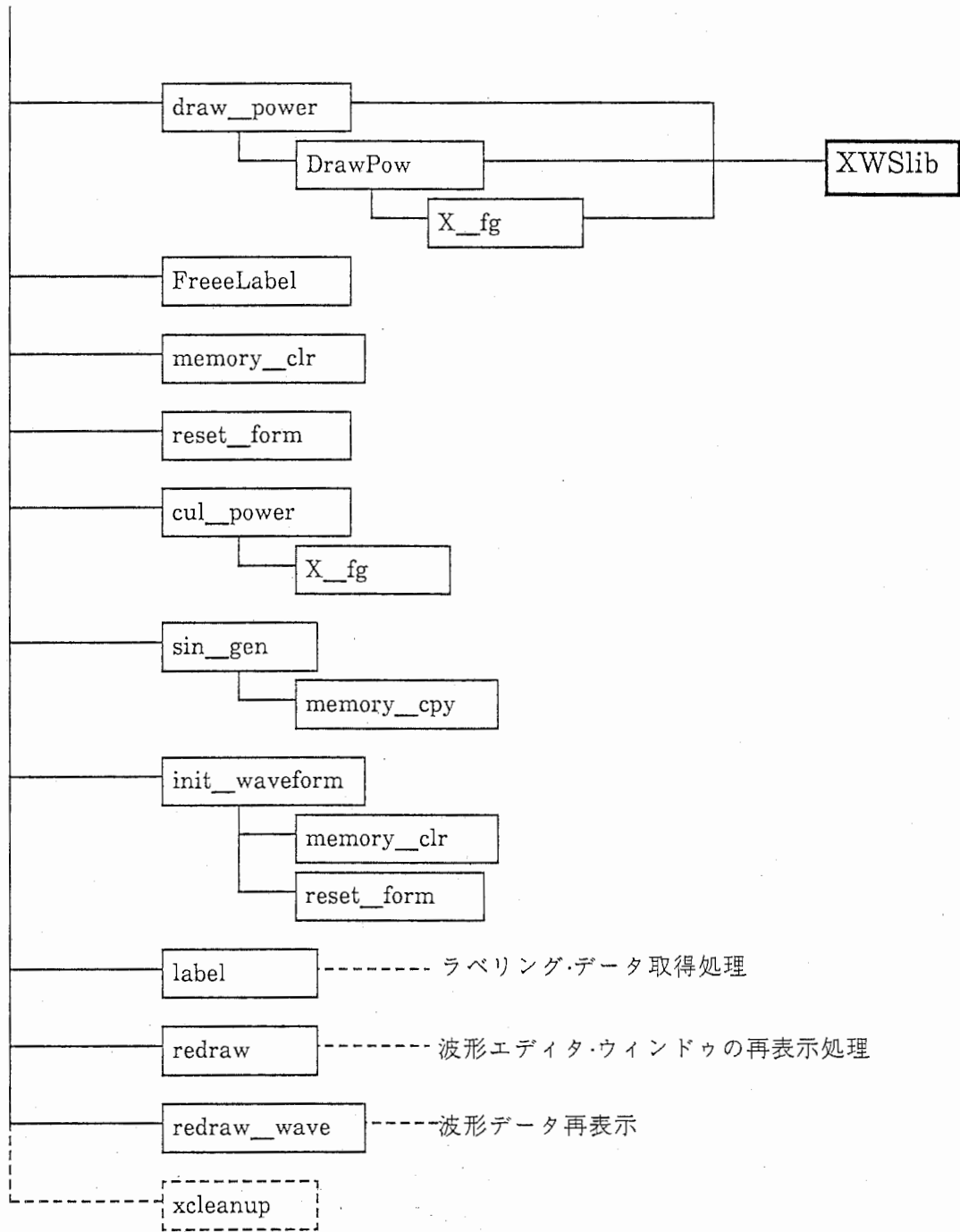




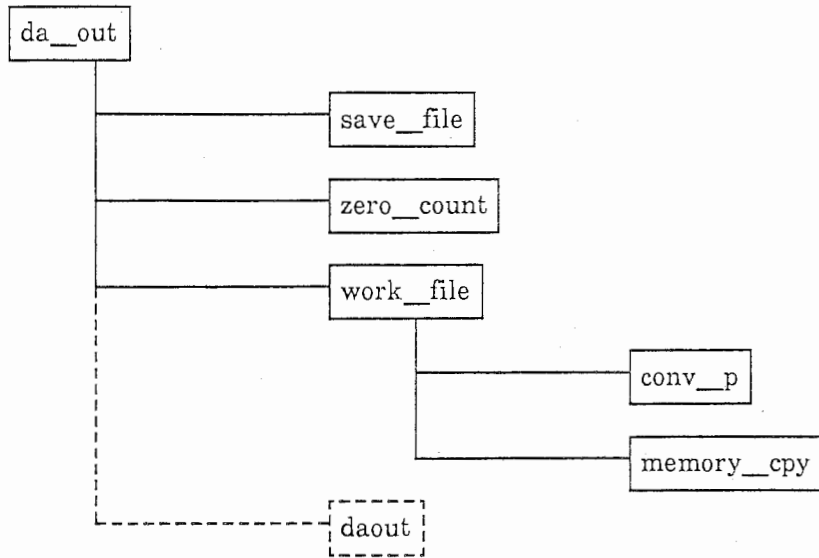




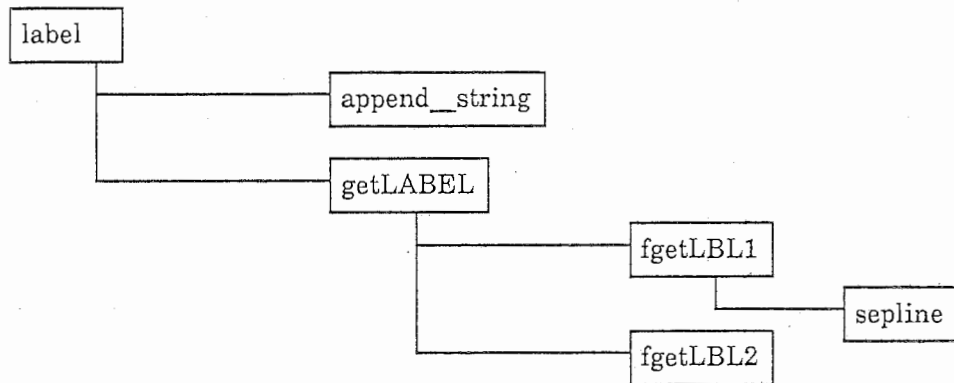




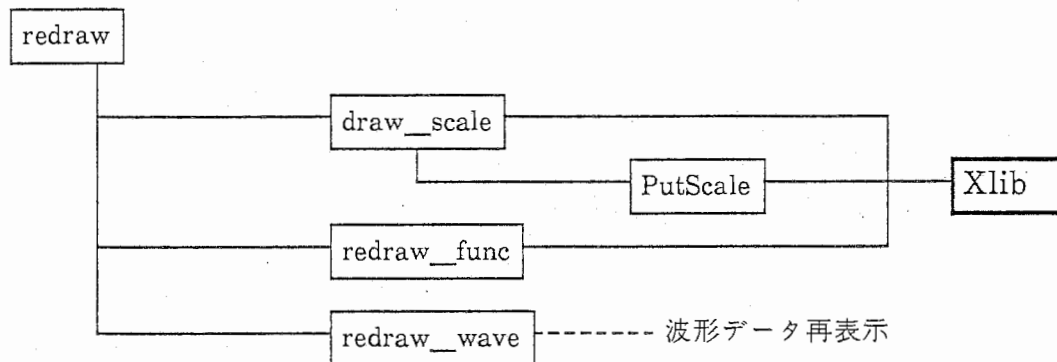
(xcleanupについては別紙参照のこと)

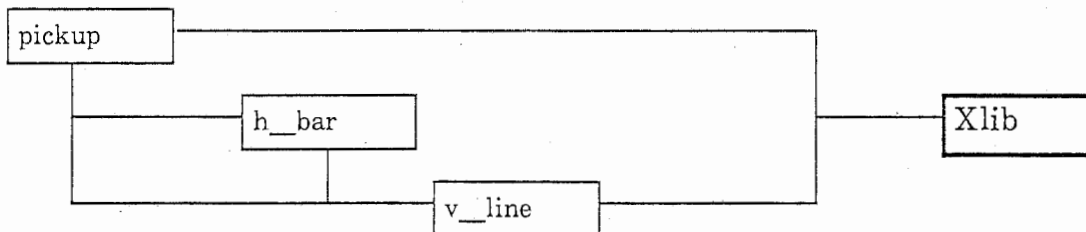
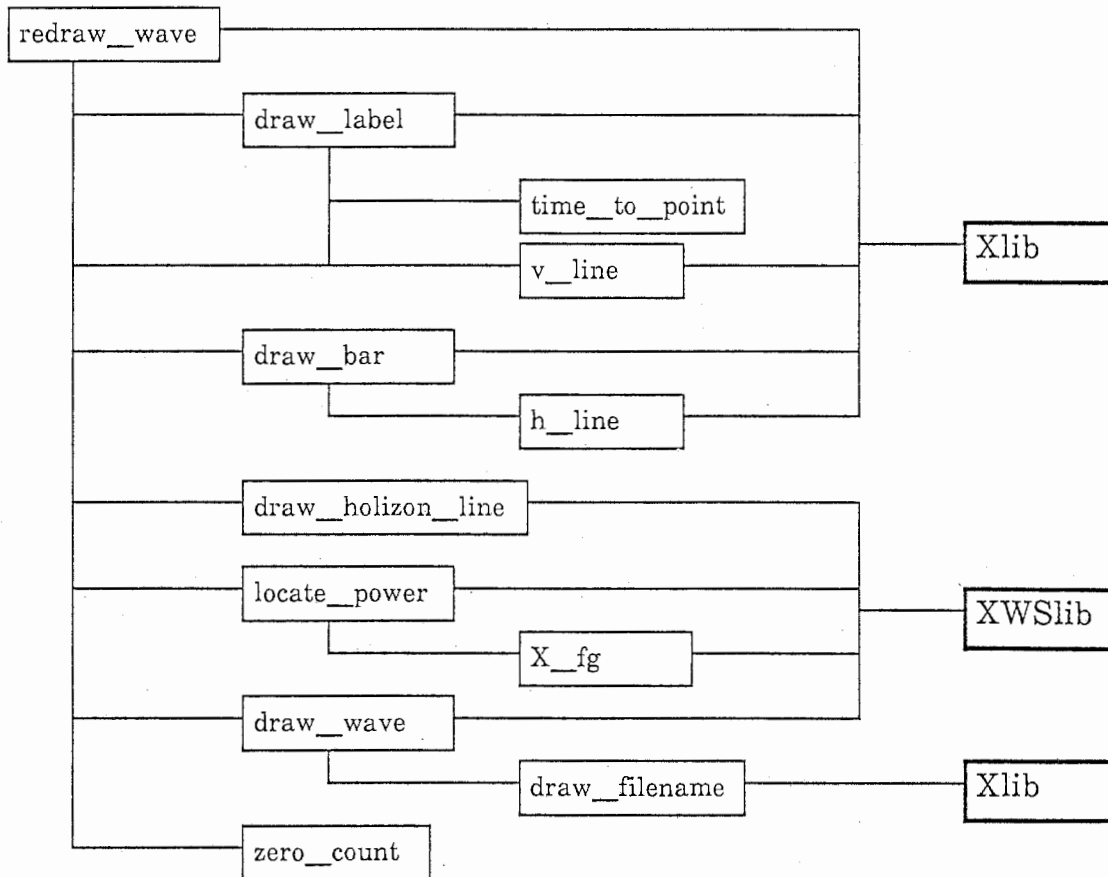


(daoutについては別紙参照のこと)



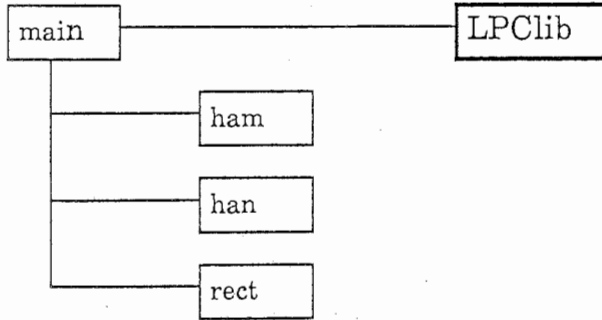
(getLABEL以下のモジュールは別紙参照)



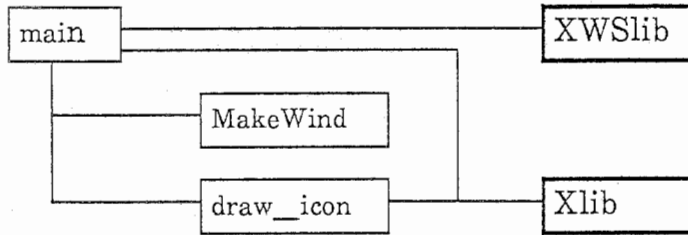


2. 相関計算(AutoCorrel)

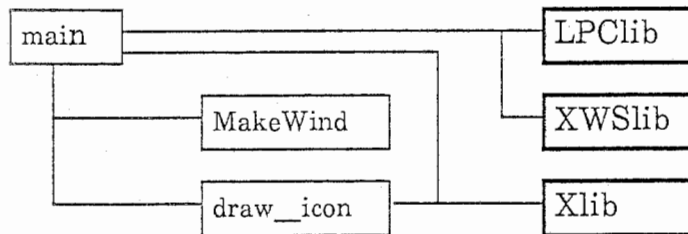
ham,han,rectは別紙参照のこと



3. ログパワー(LogPower)

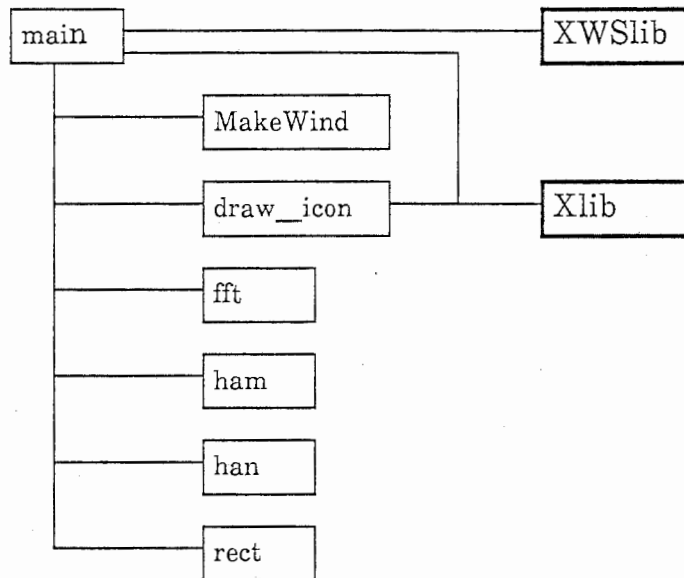


4. ランニング・スペクトログラム(RunSpect)

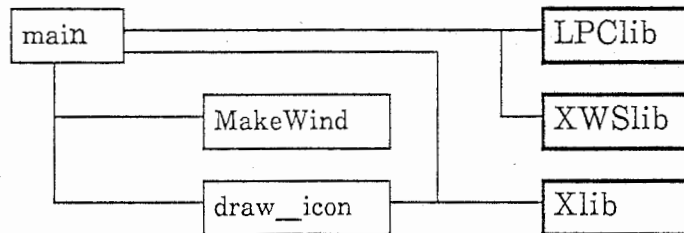


5. FFTスライス・スペクトログラム(FFTslice)

fft,ham,han,rectは別紙参照のこと



6. LPCスライス・スペクトログラム(LPCslice)



2.3 各関数の説明

1. 波形エディタの関数

(1) ADIN (ad_in.c)

- 機能

A/Dを取り込む。

- コーリングシーケンス

ADIN(buf)

buf: A/D格納エリア
(INPUT) char buf[]

(2) ad_in (ad_in.c)

- 機能

A/Dの取り込みと表示の制御を行う。
A/D格納エリアの配置を行う。

- コーリングシーケンス

ad_in(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(3) amplify (X_io.c)

- 機能

波形表示の振幅を変更する。

- コーリングシーケンス

amplify(item,y)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

y : 振幅(Y軸)のスケール値
(INPUT) short y

(4) analyze (analyze.c)

・ 機能

アナライズ・プロセスの起動制御を行う。

・ コーリングシーケンス

analyze(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(5) append_string (file_io.c)

・ 機能

第1アーギュメントに第2アーギュメントを連結した文字列を、第3アーギュメントに格納する。

・ コーリングシーケンス

append_string(str1,str2,str3)

str1 : 文字列
(INPUT) char str1[]

str1 : 文字列
(INPUT) char str2[]

str1 : 文字列
(OUTPUT) char str3[]

(6) area (area.c)

・ 機能

D/A出力及び、分析の対象とする範囲設定を行う。

・ コーリングシーケンス

area(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(7) CloseADIN (ad_in.c)

- ・ 機能

A/Dの取り込み終了処理。

- ・ コーリングシーケンス

CloseADIN()

(8) Copy__WaveForm (X_io.c)

- ・ 機能

構造体__WaveFormの複写を行う。

- ・ コーリングシーケンス

Copy__WaveForm(src,des)

src : 移す側の波形エディタのウィンドウ番号(0~2)
(INPUT) int src

des : 移される側の波形エディタのウィンドウ番号(0~2)
(INPUT) int des

(9) CreateWindows (windows.c)

- ・ 機能

波形エディタのウィンドウを生成する。

- ・ コーリングシーケンス

CreateWindows(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(10) conv__p (file_io.c)

・ 機能

デバイス座標(水平方向)をデータ・ポイント数に変換する。
(データ・ポイント数は波形開始点(左端)を0とした値)

・ コーリングシーケンス

```
int conv__p(item,point)
```

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

point : デバイス座標値
(INPUT) int point

(11) cul__power (cul__power.c)

・ 機能

音声データ・ファイルの全データをパワー放絡計算する。
その結果をファイルとする。
(/tmp/power.data%n%n : 波形エディタのウィンドウ番号)

・ コーリングシーケンス

```
cul__power(item)
```

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(12) cursor__set (cursor.c)

・ 機能

波形エディタの各ウィンドウで表示される、カーソルの定義を行う。

・ コーリングシーケンス

```
int cul__power(item)
```

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(13) cut (edit.c)

・ 機能

設定範囲の音声データを、カット・バッファに取り込む。
第2アークギュメント(flag)が2であれば、その範囲のデータを画面より消去する。

・ コーリングシーケンス

cut(item,flag)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

flag : 処理フラグ
1 データの複写
2 データの消去
(INPUT) int flag

(14) DestroyXform (X_io.c)

・ 機能

波形エディタのウィンドウに関する、xformを解除する。

・ コーリングシーケンス

DestroyXform(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(15) DrawPow (draw__power.c)

・ 機能

パワー放絡の表示を行う。

・ コーリングシーケンス

```
int DrawPow(d__mode,np,data,y__scale,y__bias,fg)
```

d__mode : 0 固定
(INPUT) int d__mode

np : データ数
(INPUT) int np

data : データ格納アレイ
(INPUT) float data[]

y__scale : データ変換定数
(dataxy__scale+y__bias)
(OUTPUT) float *y__scale

y__bias : データ変換定数
(dataxy__scale+y__bias)
(OUTPUT) float *y__bias

fg : ダミー
(INPUT) int fg

・ リターン値

正常時 : 1
異常時 : -1

(16) da_out (da_out.c)

- ・ 機能

D/A出力処理を行う。

- ・ コーリングシーケンス

da_out(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(17) deiconify (X_io.c)

- ・ 機能

波形エディタのウィンドウをアイコン化する制御を行う。

- ・ コーリングシーケンス

deiconify()

(18) direx (direx.c)

- ・ 機能

指定パスのディレクトリをメニュー化する制御を行う。
選択されたファイル名が返される。

- ・ コーリングシーケンス

char *direx(item,path)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

path : パス名
(INPUT) char *path

(19) disp_logpower (analyze.c)

・ 機能

ログ・パワー表示の制御を行う。
LogPowerを子プロセスとして、起動制御する。

・ コーリングシーケンス

disp_logpower(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(20) disp_runspect (analyze.c)

・ 機能

ランニング・スペクトログラム表示の制御を行う。
RunSpectを子プロセスとして、起動制御する。

・ コーリングシーケンス

disp_runspect(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(21) disp_spectrum (analyze.c)

・ 機能

スペクトログラム表示の制御を行う。
spect_outを子プロセスとして、起動制御する。

・ コーリングシーケンス

disp_spectrum(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(22) draw (X_io.c)

- ・ 機能

波形エディタ・ウィンドウの初期状態の表示の制御を行う。

- ・ コーリングシーケンス

draw(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(23) draw_bar (X_io.c)

- ・ 機能

波形エディタ・ウィンドウの表示ゲイン・インジケータ部を表示する。

- ・ コーリングシーケンス

draw_bar(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(24) draw_filename (X_io.c)

- ・ 機能

波形エディタ・ウィンドウに表示されている、波形データのファイル名を表示する。

- ・ コーリングシーケンス

draw_filename(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(25) draw_horizon_line (X_io.c)

- ・ 機能

波形エディタ・ウィンドウの波形表示部に水平線を引く。

- ・ コーリングシーケンス

draw_horizon_line(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(26) draw_icon (X_io.c)

- ・ 機能

波形エディタ・ウィンドウをアイコン表示にする。

- ・ コーリングシーケンス

draw_icon()

(27) draw_label (X_io.c)

・ 機能

波形エディタ・ウィンドウに表示されている、音声データのラベリング表示及び消去を行う。

・ コーリングシーケンス

draw_label(item,sw)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

sw : 処理フラグ
0 消去
1 表示
(INPUT) short sw

(28) draw_power (draw_power.c)

・ 機能

パワー放絡表示の制御を行う。

・ コーリングシーケンス

draw_power(item,sw)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

sw : ダミー
(INPUT) short sw

(29) draw__scale (X_io.c)

・ 機能

波形エディタ・ウィンドウの時間スケール表示の制御を行う。

・ コーリングシーケンス

draw__scale(item,nopnt)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

nopnt : 表示する波形データの数
(INPUT) int nopnt

(30) draw__wave (X_io.c)

・ 機能

波形及び編集ファイル名表示の制御を行う。

・ コーリングシーケンス

draw__wave(item,data,nopnt)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

data : 波形データ格納アレイ
(INPUT) short *data

nopnt : 表示する波形データの数
(INPUT) int nopnt

(31) edit (edit.c)

- ・ 機能

音声データ編集の制御を行う。

- ・ コーリングシーケンス

edit(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(32) FreeLabel (FreeLabel.c)

- ・ 機能

ラベリング表示で配置した、メモリーの解放を行う。

- ・ コーリングシーケンス

FreeLabel(label)

label : 構造体__labelのポインタ
(INPUT) LABEL *label

(33) file (file.c)

- ・ 機能

音声データ及び、ラベリング・ファイルの操作(ロード、セーブ、パス設定)の制御を行う。

- ・ コーリングシーケンス

file(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(34) flag (flag.c)

・ 機能

波形エディタ・ウィンドウのセーブ・フラグの表示及び、消去を行う。

・ コーリングシーケンス

flag(item,sw)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

sw : 処理フラグ
0 消去
1 表示
(INPUT) short sw

(35) func_sw (X_io.c)

・ 機能

波形エディタ・ウィンドウのファンクション・メニュー上での、
カーソル移動時の前景/背景の変換操作を行う。

・ コーリングシーケンス

flag(win,func,sw)

win : ウィンドウ識別子
(INPUT) Window win

func : ファンクション名
(INPUT) char *func

sw : 背景の識別子
0 黒
1 白
(INPUT) int sw

(36) gain (edit.c)

- ・ 機能

波形の振幅を変更する。

- ・ コーリングシーケンス

gain(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(37) get_default (X_io.c)

- ・ 機能

実行イメージのファイルがある、ディレクトリを得る。

- ・ コーリングシーケンス

get_default()

(38) get_event (X_io.c)

・ 機能

波形エディタのウィンドウに関する、イベント処理制御を行う。

・ コーリングシーケンス

get_event(item,cate)

item : イベントの起こった、波形エディタのウィンドウ番号
(OUTPUT) int *item

cate : マウスでクリックした、ファンクション・メニューの番号
("A/D"を0とし、右に寄る程大きい値を取る)
(OUTPUT) int *cate

(39) get_exec (X_io.c)

・ 機能

実行イメージ・ファイルの存在するディレクトリと、実行イメージの
ファイル名と連結させた、文字列のポインタを返す。

・ コーリングシーケンス

char *get_exec(name)

name : 実行イメージのファイル名
(INPUT) char *name

(40) h_bar (X_io.c)

・ 機能

アーギュメントで指定された、ウィンドウ上に指定座標の2点に垂直線をウィンドウ一杯に引く。

・ コーリングシーケンス

h_bar(wid,sx,x)

wind : ウィンドウ識別子
(INPUT) Windowwind

sx : 垂直線を引くX座標値 (DC)
(INPUT) int sx

x : 垂直線を引くX座標値 (DC)
(INPUT) int x

(41) h_line (X_io.c)

・ 機能

アーギュメントで指定された、ウィンドウの指定Y座標上に水平線を引く。

・ コーリングシーケンス

h_line(wid,y,width)

wind : ウィンドウ識別子
(INPUT) Windowwind

y : 水平線を引くY座標値 (DC)
(INPUT) int y

width : 水平線の幅 (DC)
(INPUT) int width

(42) Init_CreateWindows (windows.c)

- ・ 機能

WB起動時の、波形エディタ・ウィンドウの作成を行う。

- ・ コーリングシーケンス

Init_CreateWindows()

(43) iconify (X_io.c)

- ・ 機能

波形エディタ・ウィンドウをアイコン化にする処理。

- ・ コーリングシーケンス

iconify()

(44) init_AnIPID (file_io.c)

- ・ 機能

構造体__AnIPID(アナライザ・プロセス・テーブル)を初期化する。

- ・ コーリングシーケンス

init_AnIPID()

(45) `init_waveform` (`file_io.c`)

- ・ 機能

構造体 `_WaveForm`(指定の波形エディタ・ウィンドウ用)の初期化処理

- ・ コーリングシーケンス

`init_waveform(item)`

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(46) `init_window` (`X_io.c`)

- ・ 機能

波形エディタ・ウィンドウの初期化制御を行う。

- ・ コーリングシーケンス

`init_window()`

(47) `insert` (`edit.c`)

- ・ 機能

カット・バッファのデータを指定の波形エディタ・ウィンドウに挿入する。

- ・ コーリングシーケンス

`insert(item)`

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(48) KillChild (X_io.c)

・ 機能

指定の波形エディタ・ウィンドウに関する、アナライザ・プロセス
全てをKillする。

・ コーリングシーケンス

KillChild(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(49) label (label.c)

・ 機能

ラベル・ファイルより、データ読み込みの制御を行う。

・ コーリングシーケンス

label(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(50) label_name (file.c)

・ 機能

ラベル・ファイル名の入力処理を行う。

・ コーリングシーケンス

label_name(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(51) load (file.c)

・ 機能

指定された波形エディタ・ウィンドウに、設定されている音声データ・ファイルよりデータを読み込み、表示する。

・ コーリングシーケンス

load(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(52) load__file (file.c)

・ 機能

音声データ・ファイルを、指定のデータ数分読み込む。

・ コーリングシーケンス

load__file(fname,p,np)

fname : 音声データ・ファイル名
(INPUT) char *fname

p : データ格納アレイ
(OUTPUT) short *p

np : 読み込むデータ数
(INPUT) int np

(53) locate__power (locate__power.c)

・ 機能

パワー放絡の表示に、現在表示されている波形のスタート位置を記す。

・ コーリングシーケンス

```
locate__power(item,sw)
```

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

sw : ダミー
(INPUT) short sw

(54) MakeMenu (direx.c)

・ 機能

指定された波形エディタのウィンドウに、メニューを表示する。
選択されたメニュー(文字列のポインタ)を返す。

・ コーリングシーケンス

```
char *MakeMenu(item,n)
```

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

n : メニュー項目の数
(INPUT) int n

(55) MapSubwindows (windows.c)

- 機能

指定された波形エディタのウィンドウをマッピングする。

- コーリングシーケンス

MapSubwindows(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(56) main (main.c)

- 機能

波形エディタのメイン・モジュールである。

(57) make__correl (analyze.c)

- 機能

指定された波形エディタ・ウィンドウに表示されている、波形データの分析対象の範囲について相関計算処理を行う。
AutoCorrelを子プロセスとして起動制御する。

- コーリングシーケンス

make__correl(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(58) make_flag (flag.c)

・ 機能

波形エディタ・ウィンドウのセーブ・フラグ部に表示する、ピクセル・マップを作成する。

・ コーリングシーケンス

make_flag()

(59) make_spectrum (analyze.c)

・ 機能

指定された波形エディタ・ウィンドウに表示されている、波形データの分析対象の範囲についてスペクトラム演算処理を行う。
spect_culを子プロセスとして起動制御する。

・ コーリングシーケンス

make_spectrum(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(60) memory_clr (file_io.c)

・ 機能

指定されたメモリー(shortアレイ)を0クリアーする。

・ コーリングシーケンス

memory_clr(p,np)

p : クリアーするメモリー先頭アドレス
(OUTPUT) short *p
np : メモリークリアーする数
(INPUT) int np

(61) memory_cpy (file_io.c)

・ 機能

メモリーのコピー処理。

コーリングシーケンスのswが1の場合、p1はメモリー先頭アドレス

コーリングシーケンスのswが0の場合、p1はメモリー後尾アドレス

・ コーリングシーケンス

memory_cpy(p1,p2,np,sw)

p1 : 複写するメモリー先頭/後尾アドレス
(INPUT) short *p1

p2 : 複写されるメモリー先頭アドレス
(OUTPUT) short *p2

np : コピー回数
(INPUT) int np

sw : p1選別子 (0 or 1)
(INPUT) short sw

(62) memory__mix (file__io.c)

・ 機能

メモリーの加算処理。(p2=p1+p2)

コーリングシーケンスのswが1の場合、p1はメモリー先頭アドレス

コーリングシーケンスのswが0の場合、p1はメモリー後尾アドレス

・ コーリングシーケンス

memory__mix(p1,p2,np,sw)

p1 : 加算値が格納されている、メモリー先頭/後尾アドレス
(INPUT) short *p1

p2 : 加算結果が格納される、メモリー先頭アドレス
(OUTPUT) short *p2

np : 加算回数
(INPUT) int np

sw : p1選別子 (0 or 1)
(INPUT) short sw

(63) memory__mul (file__io.c)

・ 機能

shortアレイのデータに指定の値の掛け算を行う。(p=p*rate)

・ コーリングシーケンス

memory__mul(p,np,rate)

p : 演算結果が格納される、メモリー先頭アドレス
(OUTPUT) short *p

np : 演算回数
(INPUT) int np

rate : 掛ける値
(INPUT) float rate

(64) mix (edit.c)

・ 機能

カット・バッファのデータを、指定の波形エディタ・ウィンドウの波形データに
加算し、表示する。

・ コーリングシーケンス

mix(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(65) OpenADIN (ad_in.c)

・ 機能

A/D取り込みの為の、パラメータ設定処理

・ コーリングシーケンス

OpenADIN(rate,size)

rate : サンプリング周波数 (khz)
(INPUT) float rate

size: データバッファのサイズ
(INPUT) int size

(66) PutScale (PutScale.c)

・ 機能

波形エディタ・ウィンドウの時間スケール表示を行う。

・ コーリングシーケンス

PutScale(wind,width,height,rate,pnt,marg,start)

wind : 時間スケールのウィンドウID
(INPUT) Window wind

width : 時間スケールのウィンドウ幅 (DC)
(INPUT) long width

height : 時間スケールのウィンドウの高さ (DC)
(INPUT) long height

rate : サンプル周波数 (khz)
(INPUT) float rate

pnt : 表示の波形データ数
(INPUT) long pnt

marg : ウィンドウのボトム・マージン
(INPUT) long pnt

start : 波形データのスタート・ポイント
(INPUT) long start

(67) param (param.c)

・ 機能

指定された波形エディタ・ウィンドウに表示する、波形データ及び、アナライザに関するパラメータ設定処理を行う。

・ コーリングシーケンス

param(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(68) path (file.c)

・ 機能

読み込む、音声データ・ファイルのパスを設定する。

・ コーリングシーケンス

path(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(69) pickup (pickup.c)

・ 機能

指定ウィンドウにカーソルを表示し、マウスでクイックしたX座標値をアーギュメントに設定する。又、その位置に垂直線を引く。

・ コーリングシーケンス

pickup(wind,x,flag)

wind : ウィンドウ識別子
(INPUT) Window wind

x : X座標値
(INPUT/OUTPUT) int *x

flag: カーソル識別フラグ (1:右向き not1:左向き)
(INPUT/OUTPUT) int *x

・ リターン値

正常時 : 1

異常時 : 0

(指定以外のウィンドウで、マウスがクリックされた)

(70) power (power.c)

・ 機能

音声データが、指定レベルを満たしているか検査する。

・ コーリングシーケンス

power(data,np,lev)

data : 音声データ格納先頭アドレス
(INPUT) short *data

np : 音声データの数
(INPUT) int np

lev : レベル 0~9)
(INPUT) short lev

・ リターン値

レベルを満たす時 : 1
レベルを満たさない時 : 0

※尚、判定は100ポイント当りに換算して行っている。
又、レベルは、1データの振幅を下記のように設定している。
50 150 250 400 600 900 1200 1500 1800 2050

(71) ResizeParentWindow (windows.c)

・ 機能

波形エディタ・ウィンドウのサイズを変更する。

・ コーリングシーケンス

ResizeParentWindow()

(72) redraw (X_io.c)

- ・ 機能

波形エディタ・ウィンドウの再表示を行う。

- ・ コーリングシーケンス

redraw()

(73) redraw_func (X_io.c)

- ・ 機能

指定の波形エディタ・ウィンドウのファンクション・メニューを再表示する。

- ・ コーリングシーケンス

redraw_func(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(74) redraw_wave (X_io.c)

- ・ 機能

指定の波形エディタ・ウィンドウの波形表示部の再表示を制御する。

- ・ コーリングシーケンス

redraw_wave(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(75) reset_amp (X_io.c)

・ 機能

指定の波形エディタ・ウィンドウの波形の振幅を標準に戻す。
(表示はしない)

・ コーリングシーケンス

reset_amp(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(76) reset_form (X_io.c)

・ 機能

構造体 __WaveForm(指定の波形エディタ・ウィンドウ用)の再初期化

・ コーリングシーケンス

reset_form(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(77) reset_wdata (X_io.c)

・ 機能

指定の波形エディタ・ウィンドウに於ける、波形表示部のウィンドウ
の世界座標を標準に設定する。

・ コーリングシーケンス

reset_wdata(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(78) SelectInput (X_io.c)

・ 機能

指定の波形エディタ・ウィンドウに於ける、各ウィンドウのカーソル、
入力事象の設定を行う。

・ コーリングシーケンス

SelectInput(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(79) save (file.c)

・ 機能

指定の波形エディタ・ウィンドウに表示されている、波形データを
ファイルにセーブする処理。

・ コーリングシーケンス

save(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(80) save__file (file.c)

・ 機能

音声データを指定ファイル名で、ファイルとする。

・ コーリングシーケンス

save__file(fname,p,np)

fname : 音声データ・ファイル名
(INPUT) char *fname

p : データ格納アレイ
(OUTPUT) short *p

np : 読み込むデータ数
(INPUT) int np

(81) set__fft__slice (analyze.c)

・ 機能

FFT分析を行う、波形データの位置設定制御。
FFTスライス・スペクトログラム表示制御を行う。
FFTsliceを子プロセスとして、起動制御する。

・ コーリングシーケンス

set__fft__slice(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(82) set_lpc_slice (analyze.c)

・ 機能

LPC分析を行う、波形データの位置設定制御。
LPCスライス・スペクトログラム表示制御を行う。
LPCsliceを子プロセスとして、起動制御する。

・ コーリングシーケンス

set_lpc_slice(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(83) set_path (file_io.c)

・ 機能

WB起動コマンドより、ファイル入出力を行うデフォルト・ディレク
トリを設定する制御を行う。

・ コーリングシーケンス

set_path(n,path)

n : 第二アーギュメント(path)の数
(INPUT) int n

path : ディレクトリ名
(INPUT) char **path

(84) set_path_name (file_io.c)

・ 機能

指定波形エディタ・ウィンドウに於ける、ファイル入出力を行う
デフォルト・ディレクトリの設定を行う。

・ コーリングシーケンス

set_path_name(item,path)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

path : ディレクトリ名
(INPUT) char *path

(85) shift (file_io.c)

・ 機能

指定波形エディタ・ウィンドウに、表示されている波形をシフト表示
する処理。

・ コーリングシーケンス

shift(item,x)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

x : 波形表示の中央位置 (DC)
(INPUT) short x

(86) silence (edit.c)

・ 機能

指定波形エディタ・ウィンドウの、指定範囲の波形部をゼロとする
処理。

・ コーリングシーケンス

silence(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(87) sin_gen (sin.c)

・ 機能

正弦波のデータを生成する。

・ コーリングシーケンス

sin_gen(wave,np,rate,freq)

wave : データ格納先頭アドレス
(OUTPUT) short *wave

np : データ数
(INPUT) int np

rate : サンプルング・レート
(INPUT) float rate

freq : 周波数
(INPUT) float freq

(88) sonota (sonota.c)

・ 機能

指定波形エディタ・ウィンドウに於ける、その他の機能選択及び、
処理制御を行う。

・ コーリングシーケンス

sonota(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(89) sure (X_io.c)

・ 機能

指定波形エディタ・ウィンドウに(Sure Y/N)のメッセージを出力し、Yes/Noの照会をする。

・ コーリングシーケンス

```
int sure(item)
```

```
item : 波形エディタのウィンドウ番号(0~2)  
(INPUT) int item
```

・ リターン値

```
Yまたは、yの時 : 0  
Nまたは、nの時 : 1
```

(90) time_to_point (file_io.c)

・ 機能

指定波形エディタ・ウィンドウに表示されている、波形データに於ける時間を、ポイント数に変換して返す。

・ コーリングシーケンス

```
int time_to_point(item,time)
```

```
item : 波形エディタのウィンドウ番号(0~2)  
(INPUT) int item
```

```
time : 時間 (ms)  
(INPUT) float time
```

(91) v_line (X_io.c)

・ 機能

指定ウィンドウの指定X座標に垂直線を表示/消去する。

・ コーリングシーケンス

v_line(wind,x)

wind : ウィンドウ識別子
(INPUT) Window wind

x : X座標値 (DC)
(INPUT) int x

(92) work_file (file_io.c)

・ 機能

指定エディタ・ウィンドウに於ける、分析対象範囲の波形データ及び、
数をアーギュメントに設定する。

・ コーリングシーケンス

work_file(item,start,np)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

start : 波形データ格納先頭アドレス
(OUTPUT) short **start

np : 波形データ数格納アドレス
(OUTPUT) int *np

(93) X_fg (draw__power.c)

・ 機能

指定線の表示/消去を行う。

・ コーリングシーケンス

X_fg(np,x,y,fg)

np : 線データの数
(INPUT) int np

x : データ格納先頭アドレス [X座標値(WC)]
(INPUT) float *x

y : データ格納先頭アドレス [Y座標値(WC)]
(INPUT) float *y

fg : ダミー
(INPUT) int fg

(94) zero_count (file_io.c)

・ 機能

波形データの格納後尾アドレスより、データ値がゼロの数を返す。

・ コーリングシーケンス

```
int zero_count(p)
```

p : 波形データの格納後尾アドレス
(INPUT) short *p

(95) zero_cross (file_io.c)

・ 機能

指定波形データの振幅が、-50以上50以下で連続する回数
(100ポイント当りに換算)を返す。

・ コーリングシーケンス

```
int zero_cross(wave,np)
```

wave : 波形データの格納アレイ
(INPUT) short wave[]

np : 波形データの数
(INPUT) int np

(96) zombie (analyze.c)

・ 機能

WBの子プロセス(アナライザ)を、Kill(マウスの中央ボタンとシフト・キーを
押す)した時、実行される関数である。

・ コーリングシーケンス

```
zombie()
```


(97) zoom (zoom.c)

- ・ 機能

指定エディタ・ウィンドウの波形表示の拡大/縮小の制御を行う。

- ・ コーリングシーケンス

zoom(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(98) zoom__down (zoom.c)

- ・ 機能

指定範囲に、波形データ全体を出力する。

- ・ コーリングシーケンス

zoom__down(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(99) zoom__std (zoom.c)

- ・ 機能

音声データの先頭から、15,000ポイントを指定波形エディタ・ウィンドウに表示する。

- ・ コーリングシーケンス

zoom__std(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

(100) zoom__up (zoom.c)

- ・ 機能

指定範囲が、波形エディタ・ウィンドウ全体に表示される。

- ・ コーリングシーケンス

zoom__up(item)

item : 波形エディタのウィンドウ番号(0~2)
(INPUT) int item

2. アナライザの関数

ここに記述するのは、spect__cul spect__out を除いたアナライザに関する関数の説明を記す。

(spect__cul spect__outについては、別紙参照のこと)

(1) アナライザ共通関数

draw__icon (X__io.c)

前節(26)項参照のこと。

MakeWind

・ 機能

デフォルト・ウィンドウの設定をする。

("=widthXheight+x+y"の文字列の先頭アドレスを返す)

・ コーリングシーケンス

char *MakeWind(width,height,x,y)

width : ウィンドウの幅 (DC)
(INPUT) char *width

height : ウィンドウの高さ (DC)
(INPUT) char *height

x : ウィンドウの左端 (DC)
(INPUT) char *x

y : ウィンドウの上端 (DC)
(INPUT) char *y

(2) 相関計算 (AutoCorrel.c)

- 機能

指定の音声データ・ファイルの相関計算を行う。

- 実行時の必要アーギュメント

argv[1]: 音声データ・ファイル名
argv[2]: 音声データの数
argv[3]: 分析ウィンドウの種類 (ham,han,rect)
argv[4]: 分析ウィンドウの長さ
argv[5]: フレームシフト数
argv[6]: 相関計算の結果を出力するファイル名

(3) ログ・パワー出力 (Logpower.c)

- 機能

ログ・パワーの表示を行う。

- 実行時の必要アーギュメント

argv[1]: 相関値データ・ファイル名
argv[2]: 音声データの数
argv[3]: 分析ウィンドウの長さ
argv[4]: フレームシフト数
argv[5]: デフォルト表示ウィンドウの左端 (DC)
argv[6]: デフォルト表示ウィンドウの上端 (DC)
argv[7]: デフォルト表示ウィンドウの幅 (DC)
argv[8]: デフォルト表示ウィンドウの高さ (DC)

(4) ランニング・スペクトログラム出力 (RunSpect.c)

・ 機能

ランニング・スペクトログラムの表示を行う。

・ 実行時の必要アーギュメント

argv[1]: 相関値データ・ファイル名
argv[2]: LPC次数
argv[3]: フレーム数
argv[4]: デフォルト表示ウィンドウの左端 (DC)
argv[5]: デフォルト表示ウィンドウの上端 (DC)
argv[6]: デフォルト表示ウィンドウの幅 (DC)
argv[7]: デフォルト表示ウィンドウの高さ (DC)

(5) FFTスライス・スペクトログラム出力 (FFTslice.c)

・ 機能

FFTスライス・スペクトログラムの表示を行う。

・ 実行時の必要アーギュメント

argv[1]: FFTデータ・ファイル名
argv[2]: FFTレンジス
argv[3]: 分析ウィンドウの長さ
argv[4]: 分析ウィンドウの種類 (ham,han,rect)
argv[5]: デフォルト表示ウィンドウの左端 (DC)
argv[6]: デフォルト表示ウィンドウの上端 (DC)
argv[7]: デフォルト表示ウィンドウの幅 (DC)
argv[8]: デフォルト表示ウィンドウの高さ (DC)

(6) LPCスライス・スペクトログラム出力 (LPCslice.c)

・ 機能

LPCスライス・スペクトログラムの表示を行う。

・ 実行時の必要アーギュメント

argv[1]: 相関値データ・ファイル名
argv[2]: フレーム数
argv[3]: LPC次数
argv[4]: 音声データのLPC分析位置
argv[5]: デフォルト表示ウィンドウの左端 (DC)
argv[6]: デフォルト表示ウィンドウの上端 (DC)
argv[7]: デフォルト表示ウィンドウの幅 (DC)
argv[8]: デフォルト表示ウィンドウの高さ (DC)