ＡＤＳＰシリーズ

ＡＤＳＰ６７４－３２４

ソフトウェア・ユーザーズ・マニュアル

中部電機株式会社

目　　次

１．ライブラリの概要

本ライブラリには次の２つのプロダクトが含まれています。

１．ＡＤＳＰ６７４ボード（以下　ＤＳＰ）をブラックボックスの２チャンネルＦＦＴ演算ユニットとして使用するホスト用ライブラリです。ホスト用に作成したユーザプログラムとリンクし、関数呼び出しによりライブラリの初期化・データ授受・演算等の処理を行います。ＤＳＰとのハンドリングはライブラリ内で行い、全体の処理の主導権はホストプログラムが持ち、ＤＳＰはホスト側のユーザプログラムが発行するさまざまなコマンドに従って、演算処理を遂行します。

２．ＤＳＰ側のプログラムをユーザが記述する場合の演算関数ライブラリです。この場合、ＤＳＰ側のユーザプログラムが処理の主導権を持ち、先のコマンドの発行の変わりに、同等の関数呼び出しにより処理を実行します。演算のためのデータ領域等はユーザプログラムから随時直接アクセスすることが可能です。しかし、ホストの資源（ディスク、画面等）をアクセスするためには、Ｃアダプタ（別売開発支援ソフトウェア）を用いるか又は、仲立ちをするホスト側のプログラムを作成しなければなりません。

何れのプロダクトを使用しても同等に処理ができます。どちらを選択するかはホスト資源へのアクセス割合やプログラム開発効率に基づいて選択するとよいでしょう。この場合考慮に入れておかなければいけない点を以下に示します。

◎　ホスト用ライブラリにてコマンドを発行すると、ホストとＤＳＰ間でハンドリングが行われます。ＤＳＰ用ライブラリでは同等の手続きが関数呼び出しだけで済むためＤＳＰ用ライブラリを使用したほうが高速な処理ができます。

２．フォルダ構造

セットアップフォルダには、以下のファイルが収められています。

```
Windows システムフォルダ
└── A67XFFT.dll              ホスト用ダイナミックライブラリ

FFTLIB2
├── README.txt               簡単なドキュメントが記載されています。
├── SAMPLE
│   └── HOST
│       ├── sample.bas       サンプルプログラム Source File
│       ├── sample.exe       サンプルプログラム 実行ファイル
│       ├── sample.vbp       サンプルプログラム用ＶＢプロジェクト
│       └── sample.frm       サンプルプログラム用ＶＢフォーム
├── HOST
│   ├── A67XFFT.bas          ＶＢ用標準モジュール
│   ├── A67XFFT.h            Ｃ用ヘッダーファイル
│   └── A67XFFT.lib          Ｃ用インポートライブラリ
├── HELP
│   ├── A67X324.cnt          ヘルプ Contents ファイル
│   └── A67X324.hlp          ヘルプファイル
└── DSP
    ├── FFTEXEC.lib          ＤＳＰ用ライブラリ
    ├── FFTLIB67.cmd         ＤＳＰ用コマンドファイル
    ├── FFTLIB67.h           ヘッダーファイル
    ├── FFTTBL67.h           ヘッダーファイル
    ├── FFTWRK67.h           ヘッダーファイル
    ├── FFTLIB67.c           ＤＳＰ用ｍａｉｎプログラム
    └── FFTLIB67.out         ＤＳＰ用実行プログラム
```

◎　FFTLIB67.out　については、ホスト用ＤＬＬにて使用しますので、インストール後のフォルダ移動はしないでください。

３．機構詳細

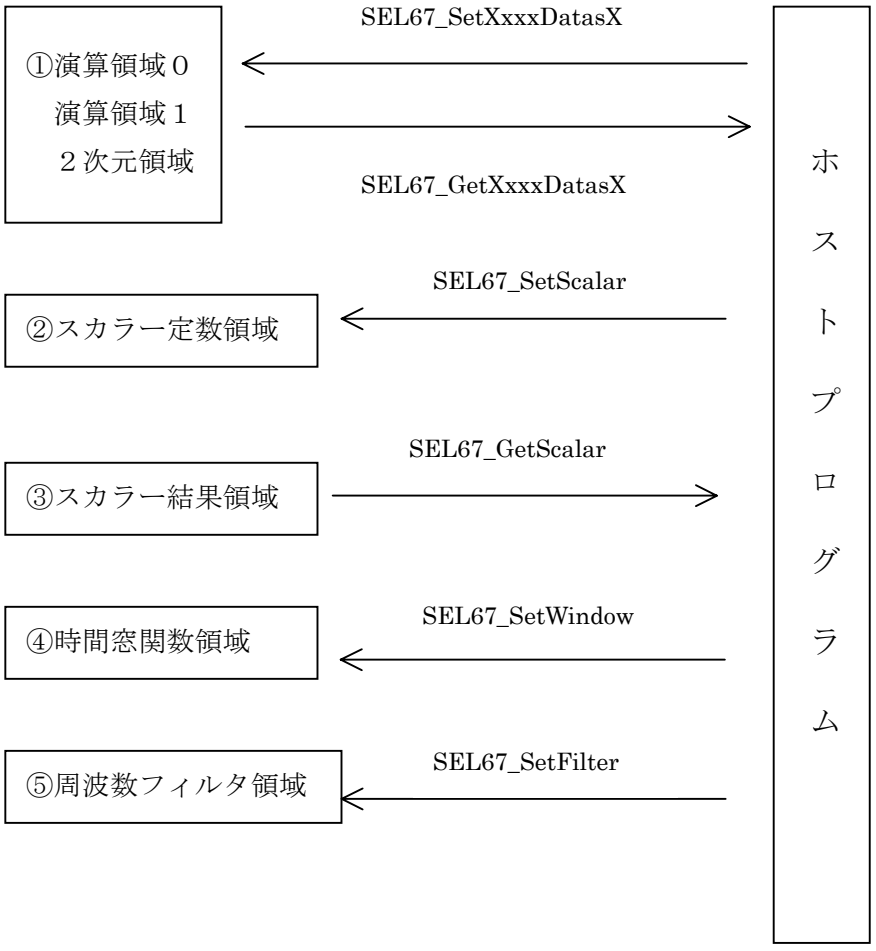
３．１　ＤＳＰメモリ配置について

１．本ライブラリは、処理実行速度を高速にするためＦＦＴ点数に応じて、ＤＳＰメモリを下表のように配置しています。

点数・次元	１６～１０２４	～２０４８	～４０９６	～１６３８４	２次元
内部メモリ (OCDMEM)	６２ KB	５０KB	３３ KB	１ KB	６２ KB
標準メモリ (SBSRAM)	１ KB	８２ KB	２３０ KB	１０５０ KB	１ KB
標準メモリ (SDRAM)	―	―	―	―	１６ MB

＊　領域確保コマンドにより、点数・次元を設定すると上記サイズが各メモリ領域の最後部に配置されますので、ＤＳＰプログラムを記述時は使用するサイズを計算のうえ、コマンドファイルにて重複しないように配置してください。

３．２　　ＤＳＰ内データブロック図



４．データ型

本ライブラリでは、データとして次の型を取り扱います。

１．スカラー型

１個の浮動小数点データで、ホストで扱う単精度浮動小数(float)と同精度です。

２．実数型

１次元の浮動小数配列です。ホストで扱う単精度浮動小数(float[])と同精度です。ホストでの double は取り扱えません。

３．複素数型

２次元の浮動小数配列です。ＦＦＴ後の周波数領域データ等はこの型になります。ホストで扱う単精度浮動小数配列(float[][2])と同精度です。

データ順は次の通りです。

data[0][0]　　先頭データの実数部

data[0][1]　先頭データの虚数部

data[1][0]　２番目データの実数部

data[1][1]　２番目データの虚数部

５．ライブラリ使用方法

５．１　　ホスト用ライブラリ

５．１．１　使用方法

ライブラリを構成している関数は、初期化関数、データ授受関数、コマンド発行関数、及び、これらの関数を組み合わせた複合関数等です。これらの関数を適宜組合せて必要な計算を順次行います。実行はおよそ次の順になります。

１．初期化関数 SEL67_Libinit() をライブラリ関数を利用するに当たってまず最初に行わなければいけません。

２．ライブラリにて使用する領域をSEL67_Fftinit()にて確保します。

３．計算の対象となるデータを、ベクトルデータ授受関数を使用してＤＳＰへダウンロードします。データは、単精度実数、単精度複素数のいずれかのデータで、ホストのメモリに一旦蓄えられたものをＤＳＰとの間で授受します。

４．計算の種類によっては係数が必要になりますから、係数ロード関数を使用してＤＳＰへダウンロードします。例えばＦＦＴ演算の時間窓関数とか、フィルタリングの係数等がそれに相当します。

５．データと係数がそろったらコマンド発行関数で実行したい計算をＤＳＰに命令します。コマンドは番号が予め決められており、ヘッダファイルfftlib67.hに定義されています。

６．計算結果はベクトルデータ授受関数でＤＳＰからホストのメモリに取り出せます。

７．処理の最後にA67X_libexit()にてＤＳＰの解放を行います。この関数を実行せずに終了すると、次回初期化関数が異常になります。

５．１．２　実装方法

本ライブラリは、Ｖｉｓｕａｌ　Ｂａｓｉｃ，Ｖｉｓｕａｌ　Ｃ＋＋から利用することができます。

１．Ｖｉｓｕａｌ　Ｂａｓｉｃにて利用する場合

メニューの「プロジェクト」→「標準モジュールの追加」を選択し、「既存のファイル」タグで”A67XFFT.BAS”（FFTライブラリⅡ）と”A67XDLL.BAS”（ADSP674標準）を追加します。

２．Ｖｉｓｕａｌ　Ｃ＋＋にて利用する場合

本ライブラリーを使用するソースファイルの先頭で#include命令を使用して、”A67XFFT.H”（FFTライブラリⅡ）・”A67XDLL.H”（ADSP674標準）ファイルを取り込みます。

又、”A67XFFT.LIB”（FFTライブラリⅡ）・”A67X32.LIB”（ADSP674標準）をインポートライブラリとしてプロジェクトへ追加します。

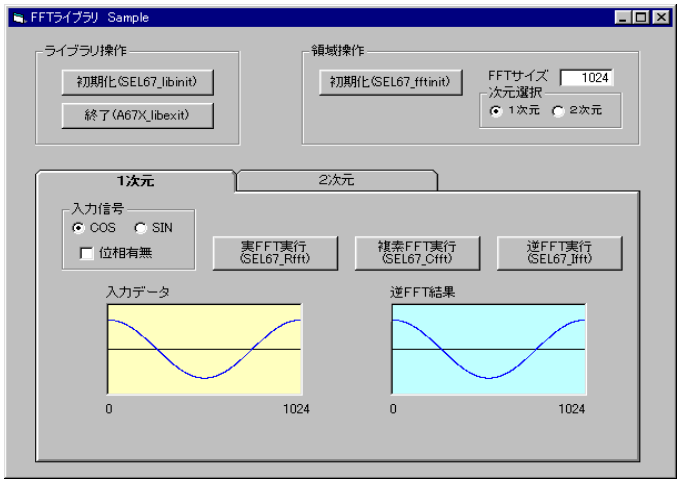
５．１．３　サンプルプログラム

本ライブラリには、Ｖｉｓｕａｌ　Ｂａｓｉｃ　Ｖｅｒ６．０のサンプルプログラムが添付されています。（ＤＳＰボード番号ゼロ番設定にして使用してください）

＜操作手順＞

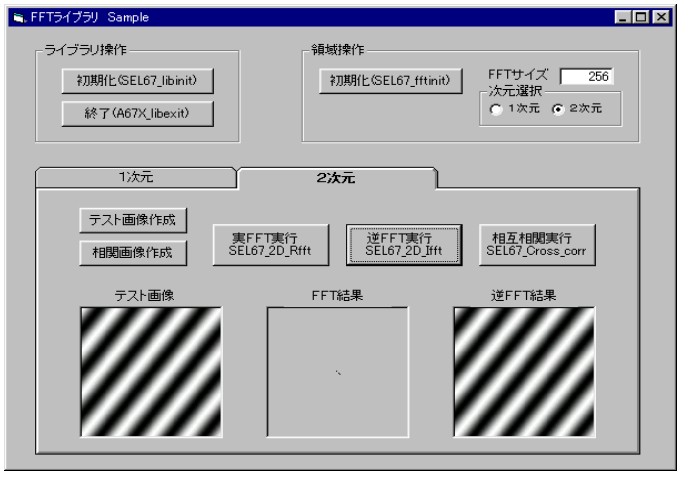
１．　　ライブラリの初期化(SEL67_libinit)

２．領域の初期化(SEL67_fftinit)



３．１次元

入力信号を選択し、「実ＦＦＴ実行」又は「複素ＦＦＴ実行」を実行する。入力データが表示されます。次に「逆ＦＦＴ実行」を実行します。逆ＦＦＴ結果が表示されます。入力信号「位相有無」をチェックし「実ＦＦＴ実行」を実行すると、逆ＦＦＴの結果位相分が無視されます。



４．２次元

「テスト画像作成」にて画像作成し、「実ＦＦＴ実行」又は「相関画像作成」にて相関基準画像を作成し「相互相関実行」を実行します。ＦＦＴ結果もしくは相関結果が表示されます。次に「逆ＦＦＴ実行」を実行します。逆ＦＦＴ結果が表示されます。（元の画像に戻ります）

５．次元又はＦＦＴサイズ変更時は、領域確報(SEL67_fftinit)を再度実行します。

６．すべて処理が終了したら、ライブラリ操作「終了」（A67X_libexit）を実行します。

５．２　　ＤＳＰ用ライブラリ

５．２．１　使用方法

出荷時のＤＳＰ用ライブラリは、ホスト用ライブラリのＤＳＰ側コアプログラムと全く同じ内容となっています。ライブラリとして使用するには、fftlib67.cよりfftlib67u.cとしてユーザプログラムを作成してください。出荷時のmain()は必要な初期化を全て行った後、for(;;)による無限ループを回っています。これがＤＳＰ側コアプログラムのアイドルループです。ホスト側ライブラリからのコマンドを待機し、その番号の正当性をチェックした後、番号に応じた処理ルーチンへ分岐テーブルを使用して分岐し、処理結果の真偽でエラーの有無を判定し、その旨をホストへレポートするようになっています。

（fftlib67u.outはインストールフォルダ[fftlib2]¥DSP へ作成してください）

５．２．２　データ領域のアクセス

ホスト側ライブラリのうち、データ授受関数に相当する関数はありません。データは演算領域に直接アクセスします。但し、以下に示す全てのポインタは処理関数からアクセスしますので、あくまでも間接参照として使用し、ポインタ自体を変更してはいけません。

１．１次元演算領域は、２つあります。大域変数　float　*data0;　と float　*data1;　がその先頭番地を指標しています。このポインタを間接参照してアクセスします。例えば演算領域１の０番目と１番目に複素数ａ＋ｂｊ，ｃ＋ｄｊ（注ｊは虚数単位）を代入する場合、

```
data0[0] = a;
data0[1] = b;
data0[2] = c;
data0[3] = d;
```

とします。

２．２次元演算領域は、大域変数　float　*data_2d　がその先頭番地を指標しています。

★　上記１、２の演算領域にあるデータは、その有効な点数と次元を別の変数で管理しています。データ点数は、大域変数　m_fft->size　が管理しています。データ点数を変更する場合は併せてこれを変更します。データ次元は、大域変数 m_fft->dim が管理しています。これらの変数は２つの演算領域共通です。

３．その他の領域

窓係数データ	float *windata
フィルタ係数	float *fildata
定数スカラ領域	float　cnsscl
出力スカラ領域	float　outscl

５．２．３　コマンドの実行

ホスト側ライブラリのうち、コマンド発行関数は分岐テーブルを使った分岐に相当します。例えば、ホスト側ライブラリで

```
long     rslt;
rslt = SEL67_Command(CFFT);
```

は、ＤＳＰ側ライブラリでも次の様なマクロ定義がされているため同様の記述ができます。

```
int      rslt;
#define  SEL67_Command(cmd)          (*function[cmd])()
rslt = SEL67_Command(ＣFFT);
```

この分岐テーブルによる関数呼び出しは、分岐テーブル本体が記述されているプログラムとコマンド番号定義　FFTLIB67.H　との間につじつまが合わせてあります。例えば、CFFTコマンドを実行したい場合、ホスト側ライブラリで、SEL67_Command(CFFT)を実行すると、コマンド番号としてFFTLIB67.Hで定義されている　３　がＤＳＰに渡ります。ＤＳＰ側ではこれを受けて、function[]の　３番目に設定されている関数(この場合ＣFFTの処理をする為の複素ＦＦＴ関数が呼び出されます。

５．２．４　複合コマンドの実装

ＤＳＰ側ライブラリには、ホスト側ライブラリの複合コマンドに相当する関数は用意されていません。複合関数は、データ授受関数とコマンド発行関数の組合せで実現されています。データ授受はポインタを用いて直接アクセスし、コマンド発行は関数を直接呼出してください。

６．ライブラリ関数の説明

６．１　　関数一覧

初期化関数

long SEL67_Libinit();	ライブラリ初期化
long SEL67_Fftinit();	FFT領域設定

データ授受関数

long SEL67_SetRealDatas();	実数データ書き込み
long SEL67_GetRealDatas();	実数データ読み出し
long SEL67_SetCplxDatas1,1S,2,2S();	複素数データ書き込み
long SEL67_GetCplxDatas1,1S,2,2S();	複素数データ読みだし
long SEL67_SetScalar();	スカラデータ書き込み
long SEL67_GetScaler();	スカラデータ読みだし
long SEL67_SetWindow();	窓関数データセット
long SEL67_EjectWindow();	窓関数データ破棄
long SEL67_SetFilter();	フィルタデータセット

コマンド発行関数

long SEL67_Command();	コマンド発行

複合関数

long SEL67_Rfft();	実数ＦＦＴ
long SEL67_Ifft();	逆ＦＦＴ
long SEL67_Cfft();	複素数ＦＦＴ
long SEL67_2D_Rfft();	２次元実数ＦＦＴ
long SEL67_2D_Ifft();	２次元逆ＦＦＴ
long SEL67_2D_Cfft();	２次元複素数ＦＦＴ
long SEL67_2D_Cross_corr();	２次元相互相関
long SEL67_R_Power();	実数データのパワー＃１
long SEL67_R_Logpower();	実数データのパワー＃２
long SEL67_C_Power();	複素数データのパワー＃１
long SEL67_C_Logpower();	複素数データのパワー＃２
long SEL67_PeakScan();	パワー最大値検出
long SEL67_OverAll();	パワーオーバーオール
long SEL67_PoverAll();	パワーパーシャルオーバーオール

注：本ライブラリで示した複合関数はＤＳＰのＦＦＴライブラリⅡで実行可能な処理のほんの一部です。

６．２　　関数詳細

関数名　ＦＦＴライブラリ初期化関数

定義　long SEL67_Libinit(bdno,type)

引数　long bdno;　　DSPボード番号

long type;　　DSPプログラムタイプ(0:標準,1:ユーザ仕様)

戻り値　0 正常です。

1 ライブラリ初期化エラー

2 ボード実装エラー

3 プログラム実行エラー

4 ボード番号エラー

5 プログラムロードエラー

備考　異常の原因としては次の事が考えられます。

●”インストールフォルダ¥DSP”にＤＳＰプログラム”fftlib67(u).out”が存在しない。

●bdnoで指定した番号のボードが接続されていない。

●ボードを拡張ラックに実装している場合、拡張ラックが正常に作動できる状況になっていない。（電源、ケーブル接続等）

●DSPボードの異常

参考　DSPプログラムタイプ　０　：　fftlib67.out,　１　：　fftlib67u.out　が実行されます。

関数名　ライブラリ領域確保関数

定義　long SEL67_Fftinit(n, dim)

引数　long n;　　FFTサイズ

long dim;　　FFT次元

戻り値　0　　正常です。

0以外　　異常です。

説明　FFT演算用領域の確保、初期処理を行います。

関数名　実数・複素数データダウンロード関数

定義　long SEL67_SetRealDatas(bfnum,len,data);　実数

long SEL67_SetCplxDatas1(bfnum,len,data);　複素数1

long SEL67_SetCplxDatas1S(bfnum,len,data);　複素数2

long SEL67_SetCplxDatas2(bfnum,len,real,imag);　複素数3

long SEL67_SetCplxDatas2S(bfnum,len,real,imag);　複素数4

引数　long bfnum;　演算領域番号 (0/1/2)

(領域番号2 ： ２次元データ領域)

long len;　データ数

float *data;　データアドレス

float *real;　データアドレス・実数部

float *imag　データアドレス・虚数部

戻り値　0　正常です。

0以外　異常です。

説明　データ数は実数・複素数におけるデータ数です。次のデータとして扱われます。実数データ　float data[len];

複素数データ１　float data[len][2];

複素数データ３，４　float real[len];

float imag[len];

バッファ番号を直接指定しますから、ＤＳＰ内の対象演算領域とは無関係に引数bufnumで指定された演算領域をアクセスします。

複素数１，３は、データが光学配列とみなしダウンロード時に並び替えをします。

関数名　実数・複素数データ・アップロード関数

定義　long SEL67_GetRealDatas(bfnum,len,data);　実数

long SEL67_GetCplxDatas1(bfnum,len,data);　複素数1

long SEL67_GetCplxDatas1S(bfnum,len,data);　複素数2

long SEL67_GetCplxDatas2(bfnum,len,real,imag);　複素数3

long SEL67_GetCplxDatas2S(bfnum,len,real,imag);　複素数4

引数　long bfnum;　演算領域番号 (0/1/2)

(領域番号2 ： ２次元データ領域)

long len;　データ数

float *data;　データアドレス

float *real;　データアドレス・実数部

float *imag　データアドレス・虚数部

戻り値　0　正常です。

0以外　異常です。

説明　データ数は実数・複素数におけるデータ数です。次のデータとして扱われます。実数データ　float data[len];

複素数データ１，２　float data[len][2];

複素数データ３，４　float real[len];

float imag[len];

バッファ番号を直接指定しますから、ＤＳＰ内の対象演算領域とは無関係に引数bufnumで指定された演算領域をアクセスします。

複素数２，４は、データが光学配列とみなしアップロード時に並び替えをします。

関数名　スカラデータダウンロード関数

定義　long SEL67_SetScalar(scalar);

引数　float scalar;

戻り値　0　正常です。

0以外　異常です。

説明　スカラー値をＤＳＰにロードし、利用可能にします。

関数名　スカラデータアップロード関数

定義　long SEL67_GetScaler(scalar);

引数　float *scalar;

戻り値　0　正常です。

0以外　異常です。

説明　スカラーの結果が返るコマンドを行った後、結果を取り出すのに使用します。

関数名　時間窓関数ダウンロード関数

定義　long SEL67_SetWindow(window);

引数　float window[fftサイズ];　窓関数データ

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　ＦＦＴの時間窓関数をセットし使用可能にします。

窓関数データはＤＳＰにダウンロードされ、振幅補正係数を計算し窓関数処理に備えます。一旦この関数を行うと以後、時間窓関数の禁止関数SEL67_EjectWindow()又は、コマンド関数でWINDOW_EJECTコマンドを発行するまでは、毎ＦＦＴ演算に先立ち窓関数処理が行われます。

窓関数データはＦＦＴの点数に合った点数のデータを用意してください。レクタンギュラ窓等の場合、点数の不足分は０で埋めて２のＮ乗点になるようにしてく　ださい。

関数名　時間窓関数禁止

定義　long SEL67_EjectWindow();

引数　なし

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　ＦＦＴ演算における時間窓関数処理を禁止します。先にSEL67_SetWindow()でセットした窓関数は破棄されます。

関数名　周波数フィルタダウンロード関数

定義　long SEL67_SetFilter(filter);

引数　float filter[fftサイズ/2+1];　　周波数特性データ

戻り値　0　　正常です。

0以外　異常です。計算が続行できません。

説明　フィルタリングコマンドで周波数領域データに掛けるフィルタデータをダウンロ　ードします。この関数で扱えるフィルタは実係数のフィルタデータです。ＦＦＴ　のデータ点数をＮとして、係数テーブルは先頭からＮ／２＋１点分が有効です。負の周波数領域の係数部分はＤＳＰ内で自動生成されます。データを与える必要はありません。

関数名　コマンド発行関数

定義　long SEL67_Command(command);

引数　long command;　　コマンド番号

戻り値　0　　正常です。

0以外　処理結果がエラーを示しています。

説明　コード化されたコマンドをＤＳＰに渡し、演算等の処理を起動します。コードはヘッダファイル(FFTLIB67.H)に定数宣言されていますので、これをインクルードしてください。各コマンド名の詳細は　コマンドの説明　の章を参照してください。

関数名　実数ＦＦＴ関数

定義　long SEL67_Rfft(data);

引数　float data[];　　実数入力兼複素出力データ

戻り値　0　　正常です。

0以外　異常です。計算が続行できません。

説明　実数データに対してＦＦＴを行い、結果を取り出します。

入出力データは次の配置になります。

入力時データ data[n];　　n番目のデータ

出力時データ data[n*2];　　n番目の実数部

出力時データ data[n*2+1];　　n番目の虚数部

データ授受関数とコマンド発行関数を組み合わせて作られた関数です。

関数名　逆ＦＦＴ関数

定義　long SEL67_Ifft(data);

引数　float data[];　入・出力データ

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　周波数データに対して逆ＦＦＴを行い、結果を取り出します。

入出力データは次の配置になります。

入出力時データ real[n*2];　n番目の実数部

入出力時データ real[n*2+1];　n番目の虚数部

データ授受関数とコマンド発行関数を組み合わせて作られた関数です。

関数名　複素数ＦＦＴ関数

定義　long SEL67_Cfft(data);

引数　float data[];　複素数入力兼複素出力データ

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　複素数データに対してＦＦＴを行い、結果を取り出します。

入出力データはSEL67_Ifftと同様の配置になります。

データ授受関数とコマンド発行関数を組み合わせて作られた関数です。

関数名　２次元実数ＦＦＴ関数

定義　long SEL67_2D_Rfft(data, sort);

引数　float data[];　実数入力兼複素出力データ

long sort;　出力データ配列指定(0/1,1:光学配列)

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　実数データに対して２次元ＦＦＴを行い、結果を取り出します。

入出力データは次の配置になります。

入力時データ data[n];　n番目のデータ

出力時データ data[n*2];　n番目の実数部

出力時データ data[n*2+1];　n番目の虚数部

データ授受関数とコマンド発行関数を組み合わせて作られた関数です。

関数名　２次元逆ＦＦＴ関数

定義　long SEL67_2D_Ifft(data, sort);

引数　float data[];　入力兼出力データ

long sort;　入力データ配列指定(0/1,1:光学配列)

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　空間周波数データに対して２次元逆ＦＦＴを行い、結果を取り出します。

入出力データは次の配置になります。

入出力時データ data[n*2];　n番目の実数部

入出力時データ data[n*2+1];　n番目の虚数部

データ授受関数とコマンド発行関数を組み合わせて作られた関数です。

関数名　２次元複素数ＦＦＴ関数

定義　long SEL67_2D_Cfft(data, sort);

引数　float data[];　複素数入力兼出力データ

long sort;　出力データ配列指定(0/1,1:光学配列)

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　複素数データに対して２次元ＦＦＴを行い、結果を取り出します。

入出力データはSEL67_2D_Ifftと同様の配置になります。

データ授受関数とコマンド発行関数を組み合わせて作られた関数です。

関数名　２次元相互相関関数

定義　long SEL67_2D_Cross_corr(data1, data2, sort);

引数　float data1[];　基準となるデータの２次元ＦＦＴ結果

float data2[];　対象となるデータの２次元ＦＦＴ結果

long sort;　入力データ配列指定(0/1,1:光学配列)

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　基準データと対象データの相互相関処理を行い、結果を data2[] へ取り出します。１次元演算領域０，１のデータは失われます。

データ授受関数とコマンド発行関数を組み合わせて作られた関数です。

関数名　リニアパワスペクトル関数（１）

定義　long SEL67_R_Power(data);

引数　float data[];　入力兼出力データ

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　実数データを入力しリニアパワを求めます。

パワ値は次式で与えられます。

X[n] : パワ値

R[n] : 実数部

I[n] : 虚数部

X[0] = sqrt(R[0]*R[0]);　直流部

X[n] = sqrt(R[n]*R[n] + I[n]*I[n]);　n番目周波数

関数名　対数パワスペクトル関数（１）

定義　long SEL67_R_Logpower(data);

引数　float data[];　入力兼出力データ

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　実数データを入力し対数パワを求めます。

パワ値は次式で与えられます。

X[n] : パワ値

R[n] : 実数部

I[n] : 虚数部

X[0] = log10(R[0]);　直流部

X[n] = log10(sqrt(R[n]*R[n] + I[n]*I[n]));　n番目周波数

関数名　リニアパワスペクトル関数（２）

定義　long SEL67_C_Power(data);

引数　float data[];　入力兼出力データ

戻り値　0　正常です。

0以外　異常です。計算が続行できません。

説明　入力データが複素数であることを除いて SEL67_R_Power() と同じです。

入力時データ data[n*2];　n番目の実数部

入力時データ data[n*2+1];　n番目の虚数部

関数名　対数パワスペクトル関数（２）

定義　long SEL67_C_Logpower(data);

引数　float data[];　　入力兼出力データ

戻り値　0　　正常です。

0以外　　異常です。計算が続行できません。

説明　入力データが複素数であることを除いて SEL67_R_Logpower() と同じです。

入出力データ仕様はSEL67_C_Power同様です。

関数名　パワーオーバーオール関数

定義　long SEL67_OverAll(data, outdata);

引数　float data[];　　入力データ

float outdata;　　出力データ

戻り値　0　　正常です。

0以外　　異常です。計算が続行できません。

説明　FFTサイズ/2＋1のパワー値を入力しオーバーオール値を計算します。

関数名　パーシャルパワーオーバーオール関数

定義　long SEL67_POverAll(data, outdata, low, high);

引数　float data[];　　入力データ

float outdata;　　出力データ

long　low;　　パーシャル最小値

long　high;　　パーシャル最大値

戻り値　0　　正常です。

0以外　　異常です。計算が続行できません。

説明　FFTサイズ/2＋1のパワー値を入力し指定パーシャル範囲内でパワー値を加算します。

関数名　パワーピーク値検出関数

定義　long SEL67_PeakScan(peak, data);

引数　float data[];　　入力データ

float peak;　　出力データ

戻り値　0　　正常です。

0以外　　異常です。計算が続行できません。

説明　FFTサイズ/2＋1のＦＦＴ済データを入力し、最大値を検出します。

７．コマンドの説明

コマンド発行関数でＤＳＰに渡す命令コードには初期化命令、演算命令、ＩＯの制御命令等さまざまな命令があります。命令コードは倍精度整数で、各命令処理に対応して番号が決めてあります。コード番号はヘッダファイルに定数宣言してありますからこれを取り込んでください。この整数コードをコマンド発行関数に渡しますが、将来のバージョンアップに備えて、番号値を直接コーディングせず、ヘッダファイル(FFTLIB67.H)に定義してあるニーモニック（定数宣言）を使用してください。また、命令によっては実行する前に所定のデータがセットされていないと無意味な命令もありますので注意が必要です。

７．１　　機能別コマンド一覧

ＦＦＴ関連計算処理

INIT	ＦＦＴライブラリ初期化
END	ＦＦＴライブラリ終了
CFFT	ＦＦＴ実行
IFFT	逆ＦＦＴ実行
CFFT_2D	２次元ＦＦＴ実行
IFFT_2D	２次元逆ＦＦＴ実行
LPOWER	リニアパワから対数パワを計算する
POWER	ＦＦＴデータに対してパワを求める
POWER2	ＦＦＴデータに対して２乗パワを求める
EXECFIL	周波数軸データに対してフィルタリングを行う
OVERALL	パワのオーバオール値を求める
PART_OVER_ALL_xxxx	パワのパーシャルオーバオールを求める
PEAKSCAN	パワの最大値を検出する

ＦＦＴ応用計算処理

AUTO_CORR	ＦＦＴによる自己相関関数を求める
CROSS_CORR	ＦＦＴによる相互相関関数を求める
TRANSFER	伝達関数を求める
CTFR_TO_BODE	複素伝達関数を利得と位相に変換する
CONVO_xxxx	ＦＦＴによるコンボリュージョンを行う
DECONVO_xxxx	ＦＦＴによるデコンボリュージョンを行う
COHERENCY_xxxx	コヒーレンス関数を求める

データ型変換と補助処理

SETWIN	窓関数を使用可能にする
EJTWIN	窓関数を無効にする
SETFIL	周波数フィルタデータをセットアップする

データ領域選択

SEL0	演算領域０を選択する
SEL1	演算領域１を選択する

（注：コマンド名中の　xxxx　等は、その部分が異なる複数のコマンドがある事を示しています。）

７．２　　命令コード詳細

この章は各種コマンドの意味や使用方法をコマンドごとに詳しく述べています。
説明は次の書式に従っています。

命令コード名	コマンドニーモニックです。ヘッダファイルに定数定義されている名称です。
機能要項	処理の内容を述べています。
実行条件	処理に先立ち、セットされていなければいけない条件を述べています。
実行結果	処理結果を述べています。
備考	コマンドにより特に補足すべき内容が有る場合は備考として述べています。

命令コード名	AUTO_CORR
機能要項	自己相関関数を計算します。
実行条件	対象演算領域　＝　周波数領域データ
実行結果	対象演算領域　＝　自己相関データ（複素数） 正規化された自己相関関数が得られます。虚数部は意味を持ちません。

命令コード名　　CFFT

機能要項　　対象演算領域の複素数データに対してＦＦＴを行います。SETWINで窓関数が有効になっている場合はＦＦＴに先立ち窓関数を対象演算領域に掛けます。

実行条件　　対象演算領域　＝　時系列データ（ｎ点複素数）
実数データ時は虚数部へゼロをセットしてください。

実行結果　　時間窓関数が有効の場合まず時間窓を掛けた後ＦＦＴが行われ、対象演算領域に複素周波数領域データが残ります。窓関数が有効の場合は窓関数による振幅補正も行われます。負の周波数領域データも取り出しができます。

data[0][0] ：　直流分
data[1][0] ：　基本周波数実数部
data[1][1] ：　基本周波数虚数部
data[2][0] ：　第２周波数実数部
data[2][1] ：　第２周波数虚数部
・・・・・
data[n/2][0] ：　ナイキスト周波数実数部
data[n/2][1] ：　ナイキスト周波数実数部
・・・・・
data[n-2][0] ：　第２周波数実数部（負の周波数）
data[n-2][1] ：　第２周波数虚数部（負の周波数）
data[n-1][0] ：　基本周波数実数部（負の周波数）
data[n-1][1] ：　基本周波数虚数部（負の周波数）

参考　　SETWIN, EJTWIN

命令コード名　　CFFT_2D

機能要項　　２次元演算領域の複素数データに対してＦＦＴを行います。SETWINで窓関数が有効になっている場合はＦＦＴに先立ち窓関数を対象演算領域に掛けます。

実行条件　　２次元演算領域(data_2d)　＝　データ（ｎ点×ｎ点）
実数データ時は虚数部へゼロをセットしてください。

実行結果　　時間窓関数が有効の場合まず時間窓を掛けた後ＦＦＴが行われ、対象演算領域に複素周波数領域データが残ります。窓関数が有効の場合は窓関数による振幅補正も行われます。負の周波数領域データも取り出しができます。

命令コード名　　COHERENCY_INIT

機能要項　　コヒーレンス関数計算の初期化を行います
実行条件　　特にありません。
実行結果　　この時点では結果は出力されません。
参考　　COHERENCY_EXEC , COHERENCY_END

命令コード名　　COHERENCY_EXEC

機能要項　　コヒーレンス関数計算の反復１回分を計算します。
実行条件　　演算領域０　＝　系への入力データ（複素数）
演算領域１　＝　系からの出力データ（複素数）
実数データの時は、虚数部にゼロをセットしてください。
実行結果　　この時点では結果は出力されません。
備考　　この処理を必要な回数反復して実行します。回数が多いほどコヒーレンス関数は真の値に近づきます。この関数の実行に先立ちCOHERENCY_INITの実行が必要です。
参考　　COHERENCY_INIT , COHERENCY_END

命令コード名　　COHERENCY_END

機能要項　　コヒーレンス関数計算を完了します。
実行条件　　特に無し
実行結果　　演算領域０　＝　コヒーレンス関数（実数）
備考　　COHERENCY_EXEC　を充分な回数実行した後この処理によりコヒーレンス関数の計算が完了します。
参考　　COHERENCY_INIT , COHERENCY_EXEC

命令コード名　　CONVO_INIT

機能要項　　ＦＦＴを用いたコンボリューションを行う為の準備をします。
実行条件　　演算領域１(data1)　＝　インパルス応答（実数）
実行結果　　演算領域１(data1)　＝　インパルス応答の周波数領域データ
備考　　コンボリュージョンは次式で求められます。
Ｘ　＝　ＩＦＦＴ（ＦＦＴ（ｘ）＊ＦＦＴ（Ｉ））
ｘは計算対象データ、Ｉはインパルス応答です。この処理はインパルス応答の周波数領域表現、ＦＦＴ（Ｉ）を予め求めておく為のものです。
参考　　CONVO_EXEC

命令コード名　　CONVO_EXEC

機能要項　　ＦＦＴを用いたコンボリューションを行います。
実行条件　　演算領域０(data0)　＝　計算対象の時系列データ（実数）
演算領域１(data1)　＝　インパルス応答の周波数領域表現
実行結果　　演算領域０(data0)　＝　コンボリューション結果（実数）
演算領域１　＝　変化無し
備考　　演算領域１は CONVO_INIT によりセットします。
コンボリューションは次式で求められます。
Ｘ　＝　ＩＦＦＴ（ＦＦＴ（ｘ）＊ＦＦＴ（Ｉ））ｘは計算対象データ、Ｉはインパルス応答です。この計算により演算領域１は失われません。
CONVO_INIT は１度だけ行えばよく、CONVO_EXEC は繰り返し実行できます。
参考　　CONVO_INIT

命令コード名　　CROSS_CORR

機能要項　　相互相関関数を計算します。

実行条件　　演算領域０(data0)　＝　基準となるデータのＦＦＴ結果
　　演算領域１(data1)　＝　対象となるデータのＦＦＴ結果
実行結果　　演算領域０(data0)　＝　相互相関データ（複素数）
　　虚数部は意味を持ちません。

命令コード名　　CTFR_TO_BODE

機能要項　　複素伝達関数を利得と位相に変換します。
実行条件　　対象演算領域　＝　複素伝達関数
実行結果　　対象演算領域　＝　利得・位相伝達関数データ型としては複素数型で得られます。複素数型の実数部に利得[db]、虚数部に位相[rad]が得られます。つまり

　　data[0][0] ：　直流分利得
　　data[1][0] ：　基本周波数分利得
　　data[1][1] ：　基本周波数分位相
　　data[2][0] ：　第２周波数分利得
　　data[2][1] ：　第２周波数分位相 ・・・・・

備考　　伝達関数計算コマンド TRANSFER で計算した結果をそのまま利用できます。

命令コード名　　DECONVO_INIT

機能要項　　ＦＦＴを用いデコンボリュージョンの準備をします。
実行条件　　演算領域１(data1)　＝　インパルス応答
実行結果　　演算領域１(data1)　＝　部分式ＦＦＴ（Ｉ）$^{-1}$
備考　　デコンボリュージョンは次式で行われます。

$$X = IFFT(FFT(x) * FFT(I)^{-1})$$

ｘは計算対象データ、Ｉはインパルス応答です。この処理は、ｘに依存しない部分式　ＦＦＴ（Ｉ）$^{-1}$　を予め計算しておくものです。計算の途上で演算領域０のデータは失われます。
参考　　DECONVO_EXEC

命令コード名　　DECONVO_EXEC

機能要項　　ＦＦＴを用いデコンボリューションを実行します。
実行条件　　演算領域０(data0)　＝　計算対象データ（実数）
　　演算領域１(data1)　＝　部分式
実行結果　　演算領域０(data0)　＝　計算結果（実数）
備考　　デコンボリューションは次式で行われます。

$$X = IFFT(FFT(x) * FFT(I)^{-1})$$

ｘは計算対象データ、Ｉはインパルス応答です。部分式　ＦＦＴ(Ｉ)$^{-1}$　は、DECONVO_INIT によりセットします。この計算により演算領域１は失われません。DECONVO_INIT は１度だけ行えばよく、DECONVO_EXEC は繰り返し実行できます。
参考　　DECONVO_INIT

命令コード名　　EXECFIL

機能要項　　周波数軸データに対してフィルタリングを行います。
実行条件　　対象演算領域　　＝　周波数領域データ
　　フィルタデータ領域(fildata)　＝　フィルタデータ
実行結果　　対象演算領域　＝　フィルタリングされた周波数領域データ
参考　　SETFIL

命令コード名　　FFTINIT

機能要項　　ＦＦＴライブラリの領域確保等初期化処理をします。
実行条件　　点数（m_fft->size）・次元(m_fft->dim)を予めセットしてください。
点数は、１次元１６点～１６３８４点、２次元は１６点×１６点（１６）～１０２４点×１０２４点（１０２４）の範囲で２のべき乗を指定してください。次元は、１又は２を指定してください。

命令コード名　　IFFT

機能要項　　逆ＦＦＴを行います。
実行条件　　対象演算領域　＝　複素周波数データ
実行結果　　対象演算領域　＝　複素時間領域データ
備考　　複素周波数領域データは負の周波数領域のデータも必要です。

命令コード名　　IFFT_2D

機能要項　　２次元逆ＦＦＴを行います。
実行条件　　２次元演算領域　＝　２次元空間周波数データ
実行結果　　２次元演算領域　＝　２次元空間データ（複素数）

命令コード名　　LPOWER

機能要項　　リニアパワから対数パワを計算する
実行条件　　対象演算領域　＝　リニアパワデータ（POWERコマンドで得られた結果）
実行結果　　対数パワーデータ
　　data[n] = log10(data[n]);

命令コード名　　OVERALL

機能要項　　パワ値に対してオーバオールを求めます。
実行条件　　対象演算領域　＝　パワスペクトル
実行結果　　出力スカラ領域(outscl)　＝　パワーオーバーオール値

命令コード名　　PEAKSCAN

機能要項　　ＦＦＴデータに対し、パワスペクトルの最大値を求めます。
実行条件　　対象演算領域　＝　周波数領域データ
実行結果　　出力スカラ領域(outscl)　＝　ピークパワスペクトル

命令コード名　　POVERALL_LOW

機能要項　　パーシャルオーバオールを求める範囲の下限周波数を設定します。
実行条件　　定数領域(cnsscl)　＝　周波数値ｆlow
　　ｆlow　＝　下限周波数　／　周波数分解能
　　周波数分解能　＝　サンプル周波数　／　ＦＦＴ点数
実行結果 この時点では結果は得られません。
備考　　後に行うパシャルオーバオール値の範囲を設定するものです。
　　一旦セットすると次に変更するまでこの値は保持されます。
参考　　POVERALL_HIGH,POVERALL

命令コード名　　POVERALL_HIGH

機能要項　　パーシャルオーバオールを求める範囲の上限周波数を設定します。
実行条件　　定数領域(cnsscl)　＝　周波数値ｆhigh
　　ｆhigh　＝　上限周波数　／　周波数分解能
　　周波数分解能　＝　サンプル周波数　／　ＦＦＴ点数
実行結果　　この時点では結果は得られません。
備考　　後に行うパシャルオーバオール値の範囲を設定するものです。
　　一旦セットすると次に変更するまでこの値は保持されます。
参考　　POVERALL_LOW,POVERALL

命令コード名　　POVERALL

機能要項　　パワ値に対してオーバオールを求めます。
実行条件　　対象演算領域　＝　パワスペクトル
　　周波数範囲の上下限がセットされている事
実行結果　　出力スカラ領域(outscl)　＝　パーシャルオーバーオール値
備考　　周波数範囲の上下限周波数は、POVERALL_HIGH　及び　POVERALL_LOW
で予め設定します。
参考　　POVERALL_HIGH,POVERALL_LOW

命令コード名　　POWER

機能要項　　ＦＦＴデータに対して２乗パワを求める
実行条件　　対象演算領域　＝　周波数領域データ
実行結果　　パワスペクトラムが得られます。
　　パワ値は解方まで行われます。
　　data[n] = sqrt(real * real + imag * imag)

命令コード名　　POWER2

機能要項　　ＦＦＴデータに対して２乗パワを求めます
実行条件　　対象演算領域　＝　周波数領域データ
実行結果　　パワスペクトラムが得られます。
　　パワ値は平方を開かない値となります。
　　data[n] = real * real + imag * imag;

命令コード名　　POWER_2D

機能要項　　２次元パワスペクトルを求めます。
実行条件　　２次元データ領域(data_2d)　＝　空間周波数領域データ
実行結果　　２次元データ領域に２次元パワスペクトラムが得られます。
　　パワ値は解方まで行われます。
　　data[n] = sqrt(real * real + imag * imag)

命令コード名　　SEL0

機能要項　　対象演算領域として演算領域０を選択します。
実行結果　　以後実行されるコマンドのうち演算領域の選択の機能のあるコマンドの処理は演算領域０に対して実行されます。

命令コード名　　SEL1

機能要項　　対象演算領域として演算領域１を選択します。
実行結果　　以後実行されるコマンドのうち演算領域の選択の機能のあるコマンドの処理は演算領域１に対して実行されます。

命令コード名　　SETFIL

機能要項　　周波数フィルタデータをセットします。（負領域合成）
実行条件　　周波数フィルタデータ領域(fildata)　＝　フィルタデータ
（点数の１/２＋１点分）
実行結果　　周波数フィルタデータ領域(fildata)　＝　フィルタデータ
参考　　EXECFIL

命令コード名　　SETWIN

機能要項　　時間窓関数データを使用可能にします。
実行条件　　時間窓関数領域　＝　ホストからロードした時間窓関数
実行結果　　時間窓関数による振幅補正の係数を計算します。
この処理は、時間窓関数ロード後１度だけ行ってください。
このコマンド実行後に実行するＦＦＴコマンドは、ＦＦＴ実行前に時間窓による振幅抑制処理が行われ、ＦＦＴ後に自動的に振幅補正が行われるようになります。
参考　　EJTWIN

命令コード名　　TRANSFER

機能要項　　伝達関数を計算します。
実行条件　　演算領域０(data0)　＝　系への入力データ（実数型）
演算領域１(data1)　＝　系からの出力データ（実数型）
実行結果　　演算領域０(data0)　＝　伝達関数（複素数）
備考　　演算領域１のデータも失われます。
参考　　CTFR_TO_BODE

・本マニュアルの内容は製品の改良のため予告無しに
変更される事がありますので、ご了承下さい。

中部電機株式会社

〒440-0004　愛知県豊橋市忠興 3 丁目 2-8

TEL <0532>61-9566

FAX <0532>63-1081

URL ： http://www.chubu-el.co.jp

E-mail ：csg@chubu-el.co.jp

ADSP324-324

FFT ﾗｲﾌﾞﾗﾘⅡ(67)

ｿﾌﾄｳｪｱ・ﾕｰｻﾞｰｽﾞ・ﾏﾆｭｱﾙ

2001.02　　第 1 版発行