DASmini E500シリーズ

基本サブルーチン仕様書

for

LINUX
UNIX
Windows

作成　平成１５年5月6日
作成　平成１５年7月２５日
作成　平成１５年9月１７日
作成　平成１６年5月２４日
作成　平成１６年7月２１日

システムデザインサービス株式会社

目　次

改訂履歴

・ 平成 15 年 5 月 6 日 初版作成

・ 平成 15 年 7 月 25 日 改訂

・ 平成 15 年 9 月 17 日 inet_io_blkf（）関数説明を修正

・ 平成 16 年 5 月 24 日 サブルーチン実行基本フローの inet_io_packet()->inet_io_blkf()

の順番を inet_io_blkf()->inet_io_packet()に変更

・ 平成 16 年 7 月 21 日　frame1 の設定範囲を訂正

０～4294967295　－＞　０～ 268435455 に訂正

frame2 の設定範囲を訂正

＜プリトリガ時＞

訂正前　０ ＜ 設定値 ＜　FIFOサイズ／チャンネル数－100

訂正後　０ ＜ 設定値 ＜　（FIFOサイズ／２）／チャンネル数－100

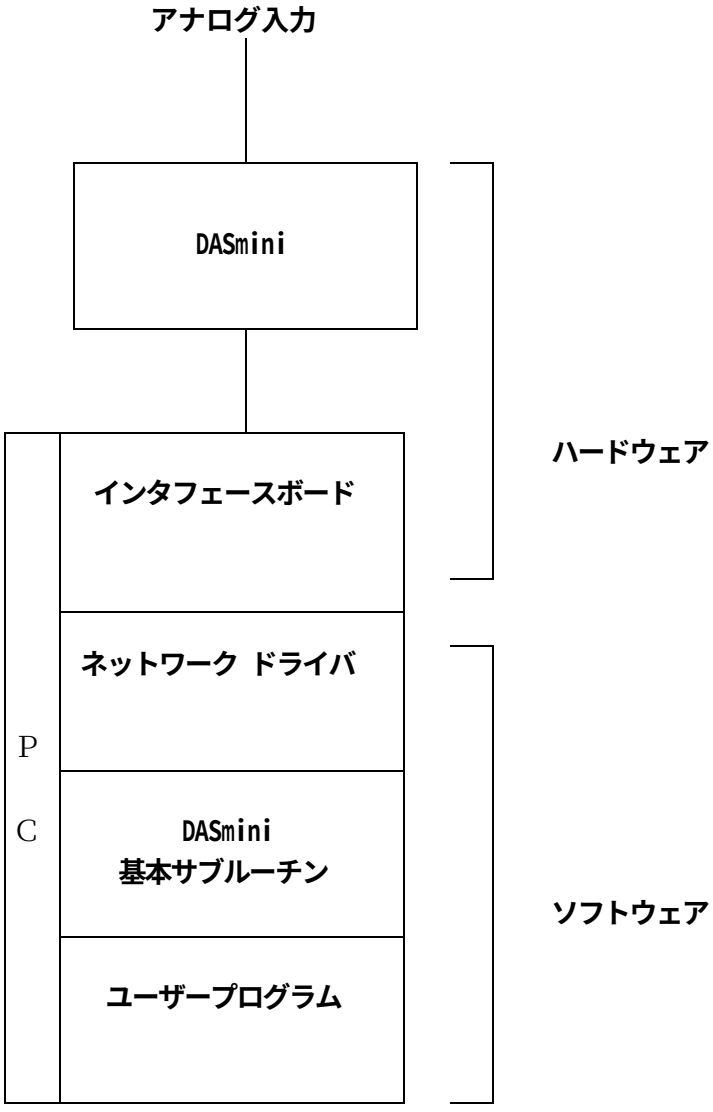
＜ポストトリガ時＞

訂正前　０ ＜ 設定値 ＜ １６７７７７２１６

訂正後　０ ＜ 設定値 ＜ １６７７７２１６

第1章 概　要

本サブルーチンは、DASmini E500シリーズをＰＣ上で使用するためのプログラムです。
本サブルーチンは、DASmini を使用するための基本操作を行うプログラムから構成されており、AD 入力及びそれらに付随する設定をサブルーチンコールによって行います。



第２章 DASmini 関数の説明

・inet_io_open()
　　DASminiと接続されているデバイスをオープンします。

・inet_io_close()
　　DASminiと接続されているデバイスをクローズします。

・inet_io_packet()
　　DASminiに対して動作を指示します。

・inet_io_cond()
　　DASminiの状態（トリガ等）を取得します。

・inet_io_stat()
　　DASminiがAD/DA動作中にエラーが発生した時、DASminiのエラーステータスを読み取ります。

・inet_io_stop()
　　DASminiの計測動作を中止させ、コマンド待ちの状態となります。
　　動作設定内容は保持されます。

・inet_io_init()
　　DASminiを初期状態に戻します。
　　動作設定内容は無効となり、新たに動作指示を行う必要があります。

・inet_io_info()
　　DASminiのインフォメーションを得ます。

・inet_io_error()
　　サブルーチン呼び出しでエラーが発生した場合、エラー内容を返します。

・inet_io_blkf()
　　データ転送のブロックサイズを設定します。

・inet_io_adstart()
　　DASminiに対してAD（入力）動作を開始させます。

・inet_io_adread()
　　DASminiからのADデータをメモリ上に読み込みます。

・inet_io_adfile()
　　DASminiからのADデータをファイルに書き込みます。

・inet_io_pre()

プリトリガが動作を行なった時、計測チャンネル数に対してのトリガ以前何データが無効データかホストに知らせます。

・inet_io_pre2()

プリトリガが動作を行なった時、１チャンネルに対してのトリガ以前何データが無効データか、及びトリガ位置のずれデータ数をホストに知らせます。

第３章 DASmini 関数の使用方法

・コントロールブロック

DASminiを制御するための環境エリアとして本関数が使用する構造体をコントロールブロックと呼びます。コントロールブロックはDASmini一台に対し、ひとつのコントロールブロックが必要になります。あらかじめ、ユーザーは構造体をstatic変数として定義する必要があります。このコントロールブロックに対するユーザーの書き込みは一切不要です。

・DASminiアーギュメント

DASminiに対して動作を指示する構造体でinet_io_packet()関数等で使用します。

内容は第４章で説明します。

コントロールブロック及びDASminiアーギュメントは、pci_sad.hファイルにその他の定義と共に納められています。ユーザーは、本関数を使用する場合pci_sad.hファイルをインクルードして使用します。

3.1 inet_io_open

機能：DASminiと接続されているデバイスをオープンします。
他の関数は、本関数呼び出し後動作可能になります。

形式：

```
int  inet_io_open(cblock,  dasarg,  ip_address);

CBLK  *cblock;

PCISAD_ARG  *dasarg;

char  *ip_address;
```

引数：

cblock	コントロールブロックのポインタ
dasarg	DASminiアーギュメントのポインタ
ip_address	Host NameかIP ADDRESSを直接指定します。 キャラクタのポインタ

戻り値：

0	正常終了
-1	異常終了

3.2 inet_io_close

機能：DASminiと接続されているデバイスをクローズします。

形式：

```
int  inet_io_close(cblock);

CBLK  *cblock;
```

引数：

cblock	コントロールブロックのポインタ

戻り値：

0	正常終了
-1	異常終了

3.3 inet_io_packet

機能：DASminiに対して動作を指示します。設定内容に関しては、第４章のDASminiアーギュメント説明を参照願います。

形式：

```
int  inet_io_packet(cblock,  dasarg);

CBLK  *cblock;

PCISAD_ARG  *dasarg;
```

引数：

cblock	コントロールブロックのポインタ
dasarg	DASminiアーギュメントのポインタ

戻り値：

0	正常終了
-1	modeの設定エラー
-2	clockの設定エラー
-4	clkmodeの設定エラー
-5	channelsの設定エラー
-6	chanum[256]の設定エラー
-7,-8	パケットコマンドエラー

3.4 inet_io_cond

機能： DASminiの状態を取得します。
このステータスはいつでも読み出すことが出来るため､ステータス情報の正当性は、inet_io_adstart()関数の呼び出し後、AD動作が完了するまでの期間となります。

形式：

```
int inet_io_cond(cblock,  data);

CBLK        *cblock;
int         *data;
```

引数：

cblock　　コントロールブロックのポインタ
data　　ステータスを格納する変数のポインタ
DASminiの状態を返します。各ビット毎に意味を持ちます。

31~16	15	14	13	12	11	10	9	8	7	6	5	4	3 ~0
*	*	*	*	*	NEM	*	*	*	EN	*	TRG	*	*

＊ 印のビットは不定データとなります。

Bit11: NEM
DASminiのFifoが空でない時に“１”となります。

Bit7: EN
DASminiが計測中に“１”となり、計測終了すると“０”となります。

Bit5: TRG
計測スタート時に“０”クリアされ、トリガを使用するモードの時に、トリガを受信すると“１”となります。
リトリガモードの場合は、１回目のトリガで“１”となりますので、２回目以降のトリガのステータスとしては使用できません。
２回目以降のトリガの判断は、NEMビットを使用し、１フレーム分のデータを読み込んでも、NEMビットが“１”の場合は次のトリガを受信して、データが格納されたと判断します。

戻り値：

0　　正常終了
-1　　異常終了

3.5 inet_io_stat

機能：DASminiに動作エラーが発生した時のステータスを取得します。
データ転送関数inet_io_adread()などの制御関数でエラーが発生した場合、inet_io_stat()でエラーの詳細を突き止めます。
尚、一度通知したエラーは通知したときにクリアされます。

形式：

```
int  inet_io_stat(cblock,  dasarg);

CBLK  *cblock;

PCISAD_ARG  *dasarg;
```

引数：

cblock	コントロールブロックのポインタ
dasarg	dasminiアーギュメントのポインタ

戻り値：

0	正常終了
-1	異常終了
8000（HEX）	SDS_PCI_FIFO_OVERFLOW
8001（HEX）	SDS_PCI_OVER_SAMPLING
8002（HEX）	SDS_PCI_ADDA_ABORT
8003（HEX）	SDS_PCI_TIMEOUT

3.6 inet_io_stop

機能：DASminiの計測を強制終了させます。設定されたパラメータは保持されますので、inet_io_packet()にて再度パラメータを設定する必要はありません。但し、DASmini内のメモリー内に取り込まれているデータは消去されます。また、この関数を実行する時はinet_io_blkf関数で指定した

形式：

```
int  inet_io_stop(cblock,  dasarg);

CBLK  *cblock;

PCISAD_ARG  *dasarg;
```

引数：

cblock	コントロールブロック	
dasarg	DASmini	アーギュメント

戻り値：

0	正常終了
-1	異常終了

3.7 inet_io_init

機能：DASminiを初期状態に戻し、設定されているパラメータは消去されます。
計測中は、強制終了しDASmini内のメモリー内に取り込まれているデータは消去されます。

形式：

```
int  inet_io_init(cblock);

CBLK  *cblock;
```

引数：

cblock	コントロールブロックのポインタ

戻り値：

0	正常終了
-1	異常終了

3.8 inet_io_info

機能：DASminiの内部情報を得ます。

形式：

```
int  inet_io_info(cblock,  statptr,  size);

CBLK  *cblock;

PCISAD_INFO  *statptr;

int  size
```

引数：

cblock	コントロールブロックのポインタ
statptr	情報データのポインタ
size	読み込む情報量 (word)) 通常はboard_idまでが有効ですので、９を設定してください。

戻り値：

0	正常終了
-1	異常終了

```
PCISAD_INFO  {
int                     fifo;
unsigned  short         fifo_wide;
unsigned  int           input_channel;
unsigned  int           output_channel;
unsigned  int           board_id;
unsigned  short         module_input;
unsigned  short         module_output;
}
```

1） fifo

DASminiに実装されているFIFOのワードサイズが入ります。。
　標準は３３５５４４３２

2） fifo_wide

DASminiに実装されているFIFOのビット長が入ります。
　現在は固定長の１６が入っています。

3） input_channel

DASminiの最大ADチャネル数が入ります。

標準は １６

4） output_channel

DASminiの最大DAチャネル数が入ります。

標準は ０

5） board_id

常に ０

6） module_input

未使用（拡張用）

7） module_output

未使用（拡張用）

3.9 inet_io_error

機能：サブルーチン呼び出しでエラーが発生した場合エラー内容を文字列で返します。

形式：

```
char  *inet_io_error(cblock);

CBLK  *cblock;
```

引数：

cblock	コントロールブロックのポインタ

戻り値：

文字列へのポインタ

3.10 inet_io_blkf

機能： １チャンネルに対する1回のinet_io_adread()のデータの転送数 （ブロック数）を指定します。inet_io_adstart()の前に、1度だけ指定してください。データ転送中に変更した場合は、保持されますが、次のinet_io_adstart()から有効となります。

形式：

```
int  inet_io_blkf(cblock,  size)

CBLK  *cblock;
int  size;
```

引数：

cblock	コントロールブロックのポインタ
size	１チャンネルに対しての、１回のブロック（データ転送）サイズを指定します。単位はワードです。 設定範囲 ： １～DASminiﾒﾓﾘｰｻｲｽﾞ／２／計測ﾁｬﾝﾈﾙ数 標準： DASminiﾒﾓﾘｰｻｲｽﾞ = 33554432（32MW） オプション： DASminiﾒﾓﾘｰｻｲｽﾞ = 67108864（64MW） 設定制限 ： ﾄﾗﾝｼﾞｪﾝﾄ計測（AD終了後、データを取り込む）で、計測を行う場合は、size×計測ﾁｬﾝﾈﾙ数が８の倍数である必要があります。

戻り値：

0	正常終了
-1	異常終了

補足： DASmini内部には、ハードウェアのメモリー及びファームウェアの中間バッファがあります。転送効率を上げるため、ファームウェアはホストからのデータ転送要求がなくても、データの先読みを行います。Sizeの設定はこの時の、データの先読みデータ量の指定も兼ねております。
よってAD動作の場合、size×計測チャンネル数指定とinet_io_adread()の読み込み数（count）が一致しない場合例１の様な転送となります。

例１ ｻﾝﾌﾟﾙｸﾛｯｸ=１Hz、size=2、計測ﾁｬﾝﾈﾙ数=10、count=10
ﾁｬﾝﾈﾙに対して２データずつ読み込みを行うため、
inet_io_adread（）を連続に行った場合に、戻り時間が約２秒、１秒以下、約２秒、１秒以下のくり返しとなる。

例２ ｻﾝﾌﾟﾙｸﾛｯｸ=１Hz、size=1、計測ﾁｬﾝﾈﾙ数=10、count=10
ﾁｬﾝﾈﾙに対して１データずつ読み込みを行うため、
inet_io_adread（）の戻り時間が約１秒のくり返しとなる。

3.11 inet_io_adstart

機能：DASminiのAD（入力）動作（計測）を開始させます。

形式：

```
int  inet_io_adstart(cblock);

CBLK  *cblock;
```

引数：

cblock	コントロールブロック

戻り値：

0	正常終了
-1	異常終了

3.12 inet_io_adread

機能：DASminiからのADデータをメモリ上に読み込みます。

形式：

```
int  inet_io_adread(cblock,buffer,count);

CBLK  *cblock;

short  *buffer;

int  count;
```

引数：

cblock	コントロールブロックのポインタ
buffer	ADデータを格納するバッファのポインタ
count	読み込むADデータ（１６ビットデータ）数 基本的には、データ転送の効率を上げるために、下記の設定にしてください。 inet_io_blkf()で設定したsize×計測チャンネル数 ただし、データ転送の効率の問題ですので、必ず一致させる必要はありません。途中及び最後等に上記と異なるサイズで転送を行っても問題ありません。

戻り値：

0	正常終了
0以外	異常終了

詳細：この関数の第 3 パラメータ count は一回に buffer に読み込むデータ数を設定します。
設定する値として 0 以外の値で blkf に関係なく自由な値が設定できます。
ただしこの関数を一回以上実行した時の count の合計数は、総データ量（frame1×channels）と一致させて下さい。
例えば総データ量が 4096 の場合、count=1000 で 4 回実行して最後に count=96 実行して終了します。
実際には DASmini とのデータ転送は blkf に合わせて内部でキューイングして行っています。
例えば総データ量が 4096(2ch,frame1=2048)で blkf=1000 の場合

DASmini から 2000 データ Read

```
inet_io_adread(,,1000)
inet_io_adread(,,1000)
```

DASmini から 2000 データ Read

inet_io_adread(,,1000)
inet_io_adread(,,1000)

DASmini から 96 データ Read

inet_io_adread(,,96)

という手順でデータ転送を行います。
尚 frame1=0 の場合は最後の DASmini からの Read は 2000 データ Read して inet_io_adread(,,96)で 96 データ buffer に渡します。

3.13 inet_io_adfile

機能：DASminiからADデータをファイルに書き込みます。

形式：

```
int  inet_io_adfile(cblock,  buffer,
                    size,  file_name,  frame);

CBLK  *cblock;

short  *buffer;

int  size;

char  *file_name;

unsigned  int    frame;
```

引数：

cblock	コントロールブロックのポインタ
buffer	テンポラリバッファのポインタ
size	テンポラリバッファのサイズ（word）
file_name	書き込むファイル名（キャラクタのポインタ）
frame	トータル書き込み数（word）

戻り値：

0	正常終了
-1	異常終了

3.14 inet_io_pre

機能：

プリトリガAD動作完了後、トリガ以前の無効データ数を得ます。
predataは設定されたプリトリガサイズに対し、トータルチャンネルにて無効であったデータ数を返します。プリトリガデータが全て有効であれば0を返します。プリトリガモード計測は、プリトリガサイズに満たないでトリガが入った場合、不足分のデータを無効データとします。また、データ転送は無効データも送られてきます。DASminiはinet_io_pre2（）関数を使用してください。

形式：

```
int  inet_io_pre(cblock,predata);

CBLK        *cblock;
int         *predata;
```

引数：

cblock	コントロールブロックのポインタ
predata	無効データ数を格納する変数のポインタ

戻り値：

0	正常終了
-1	異常終了

3.15 inet_io_pre2

機能：

プリトリガAD動作完了後、１チャンネルに対する、トリガ以前の無効データ数とトリガ位置のずれデータ数を得ます。

predataは設定されたプリトリガサイズに対し、無効であったデータ数を返します。プリトリガデータが全て有効であれば0を返します。

形式：

```
int  inet_io_pre(cblock,predata,adjust);

CBLK        *cblock;
int         *predata;
int         *adjust;
```

引数：

cblock	コントロールブロックのポインタ
predata	無効データ数を格納する変数のポインタ
adjust	ずれデータ数を格納する変数のポインタ

戻り値：

0	正常終了
-1	異常終了

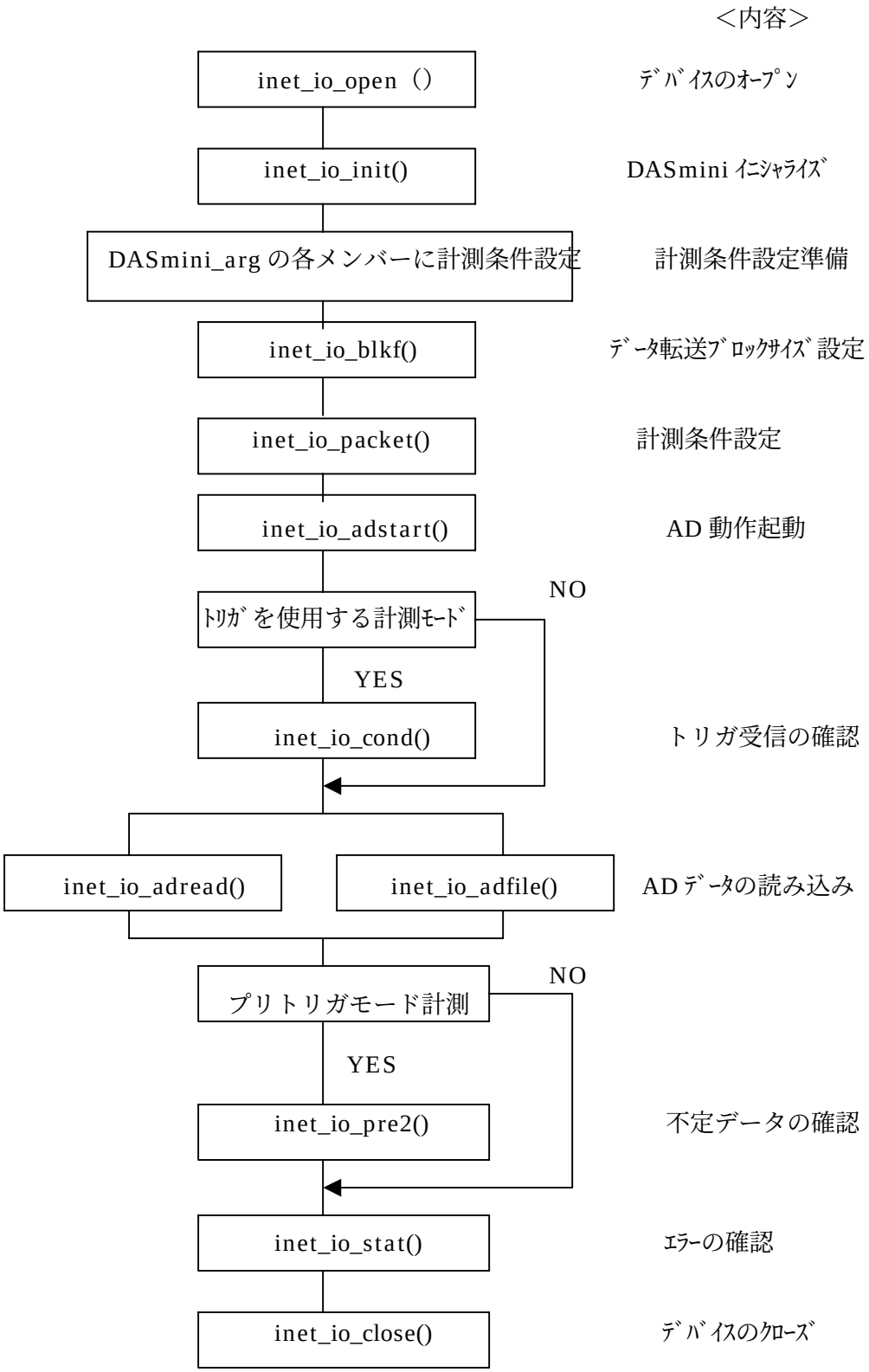
DASmini のプリトリガーAD 変換の動作はパケットで指定された presize に満たない場合でも、トリガーを検出すると有効と見なします。inet_io_adread()関数では presize 前からのデータを返します。この無効のデータ数を知るための情報です。

predataは設定されたプリトリガサイズに対し、無効であったデータ数（１チャンネル分）を返します。プリトリガデータが全て有効であれば0を返します。

また、書き込みタイミングとサンプリング周波数の関係でトリガポイントがずれる場合があります。adjustはずれたデータ量（１チャンネル分）を示します。

0=ずれ無し。n=n データ分トリガー位置が前方にずれた。presize が n 分少なくなった事を示します。よって、実際のトリガポイントのデータは設定値のポイント（presize）より、adjust 分前にある事となります。

・サブルーチン実行基本フロー



第４章 DASmini アーギュメント説明

DASminiを動作させるには、inet_io_packet関数を使用して行います。
inet_io_packet関数の第２パラメータがDASminiの動作内容になっています。

```
struct  PCISAD16_arg
        {
        unsigned  short   mode;
        unsigned  short   stat;
        unsigned  short   dastime;
        unsigned  short   attn;
        unsigned  short   gain;
        unsigned  short   trgslp;
        unsigned  short   clkmode;
         int              fifo;
         float            clock;
        int               frame1;
        int               frame2;
        int               frame3;
        int               frame4;
        int               frame5;
        unsigned  int     mutelevel;
        unsigned  short   trglevel;
        unsigned  short   trgsrc;
        unsigned  short   trgch;
        unsigned  short   channels;
        unsigned  short   chanum[256];
        unsigned  short    master;
        unsigned  short    ext_parm;
        int               dmasize;
        int               pre_mem_size;
        }PCISAD_ARG；
```

4-1　mode

DASminiに対する動作を指定します。

Define名	値（DEC）	動作
ADNORM	0	ADノントリガスタートモード
ADTRG	1	ADノーマルトリガスタートモード
ADPRE	2	ADプリトリガスタートモード
ADRET	3	ADノーマルリトリガスタートモード
ADPOST	10	ADポストトリガスタートモード
ADREPOST	11	ADポストリトリガスタートモード

4-2 stat

ステータスコマンドを実行させた時にDASminiから送られたステータスが格納されるメンバーです。

値（HEX）	ステータス
0	正常
8000	SDS_PCI_FIFO_OVERFLOW
8001	SDS_PCI_OVER_SAMPLING
8002	SDS_PCI_ADDA_ABORT
8003	SDS_PCI_TIMEOUT

4-3 dmatime

inet_io_adread,inet_io_dawrite,inet_io_adfile,inet_io_dafile時の転送タイムアウト時間を設定します。
inet_io_blkfで設定する転送数に合わせてください。
例：blkf=1000000, Sampling Clock=10000Hzの場合
blkf / Sampling Clock = 100秒以上の値を設定。

値	動作
0	基本サブルーチンのデフォルト値になります。
1～n	秒単位になります。

4-4 attn（未使用）

常に０を設定してください。（不定でも問題なし）

4-5 gain（未使用）

常に０を設定してください。（不定でも問題なし）

4-6　trgslp

各トリガモードを選択した場合有効になりトリガのスロープ指示を行います。

値	動作
0以外	立ち上がりでトリガ
0	立ち下がりでトリガ

4-7　clkmode

サンプリングクロックのソースを指定します。

Define名	値	動作
INTCLK	1	内部クロックを使用
EXTCLK	2	外部からクロックを入力する
EXTDIV	3	外部からクロックを分周する

4-8　clock

内部クロック及び外部分周を指定した場合有効で、内部クロックを指定した場合は周波数をHz単位で指定し、外部分周を指定した場合は分周値を設定します。また、内部クロックを指定した場合は、実際にサンプリング周波数として設定された値が戻されます。サンプリングクロックは内部クロックベース（8.000MHz,8.192MHz）を分周して作成します。分周の余りがでる場合は近似値が自動に設定されますので、その設定した周波数値を返します。

＜内部クロック＞
　設定範囲　　０．５Hz～最大サンプリング周波数
＜外部分周＞
　設定範囲　　２～６５５３６

4-9　frame1

AD又はDAする１チャンネルに対するデータ量を指定します。単位はワードです。但し、“０”を設定すると、無限取り込みモードとなります。また、プリトリガモードの場合はframe2の値も含んで設定してください。

設定範囲　　０～２６８４３５４５５

4-10 frame2

プリトリガ動作もしくはポストトリガ動作を指定した場合有効です。

＜プリトリガの時＞

プリトリガ動作の時は１チャンネルに対する、トリガ以前のデータ量を指定します。単位はワードです。

設定範囲　　０ ＜ 設定値 ＜　（FIFOサイズ／２）／チャンネル数－100

注）FIFOサイズ全てはプリトリガサイズに指定できません。
トリガ受信時のDASmini内部セットアップ時間が必要ですので、チャンネルあたり、100データ使用できないと考えてください。

１チャンネルに対するトリガ以後の計測データ数 ＝　frame1 － frame2
の関係になります。

＜ポストトリガの時＞

ポストトリガ動作の時は遅延サンプルクロック数を指定します。
遅延時間はサンプルクロックの指定により、変ります。
“１”の場合１サンプル遅延
設定範囲　　０ ＜ 設定値 ＜ １６７７７２１６

4-11 frame3

未使用
常に“１”を設定してください。

4-12 frame4

未使用
常に“１”を設定してください。

4-13 frame5

modeがADRETRG及びADREPOSTのときトリガ回数を指定します。
設定範囲　１～６５５３６

4-14 mutelevel

未使用。

4-15 trglevel

トリガモードを指定した場合有効でトリガがかかる電圧を指定します。
設定範囲　０～１２７（DEC）
０～７F（HEX）

値		電圧	±５V	±１０V
HEX	DEC			
0x7F	127	+FS(63/64)v	+4.92V	+9.84V
0x7E	126	+FS(62/64)v	+4.84V	+9.68V
0x41	65	+FS(1/64)v	+0.08V	+0.16V
0x40	64	0v	0V	0V
0x3F	63	-FS(1/64)v	-0.08V	-0.16V
0x01	1	-FS(63/64)v	-4.92V	-9.84V
0x00	0	-FS(64/64)v	-5V	-10V

＊）1LSB=2FS/128　入力電圧±５Vの時は　0.078125V
入力電圧±１０Vの時は 0.15625V

4-16 trgsrc

トリガを使用するモードを指定した場合有功となり、トリガのソースを指定します。

値	動作
0	AD入力チャンネルトリガ
1	外部トリガ端子

4-17 trgch

trgsrc にて、AD 入力チャンネルトリガを指定した場合有効となり、トリガとするチャンネル番号を指定します。

値	動作
1～16	AD入力チャンネル番号

4-18 channels

AD又はDAするチャンネル数を指定します。

範囲　1～256

4-19 chanum【256】

ランダムチャンネルの指定を行います。AD 又は DA するチャンネル番号を指定します。指定した順番でデータ転送が行われます。

範囲　１～２５６

chanum[０]	1 番目のチャンネル番号
chanum[１]	２番目のチャンネル番号
⋮	⋮
chanum[255]	256 番目のチャンネル番号

4-20 master,ext_parm,dmasize,pre_mem_size

未使用

DASmini－E シリーズでは、このメンバーは使用しません。

第５章 DASmini ソフトウェア作成

5-1 インストール

提供メディアからキットを適当なディレクトリにコピーしてください。

UNIX (LINUX)の場合

提供メディアからキットを適当なディレクトリにコピーしてください。

FloppyDiskの場合のコピー方法

```
mount /mnt/floppy
tar xvf /mnt/floopy/kit.tar
```

Windowsの場合

適当なディレクトリ（例えばD:\das5x0）を作成して提供メディアから\das5x0
以下のディレクトリを作成したディレクトリにコピーしてください。

5-2 キット内容

キットにはDASminiを動作させるための基本サブルーチン及びサンプル
ソフトが入っています。
ディレクトリ構成は以下のようになっています。

```
kit/
    inetlib/   sample/
```

また各ディレクトリには以下のファイルがあります。

```
inetlib/ ・・・基本サブルーチンディレクトリ
  inet_sad_lib.c   ・・・基本サブルーチンソースファイル
  pci_sad.h        ・・・基本サブルーチンインクルードファイル
  pci_sad_reg.h    ・・・基本サブルーチンインクルードファイル
  sadtp.h          ・・・ユーザープログラムインクルードファイル

sample/ ・・・サンプルソフトディレクトリ
  Makefile         ・・・UNIX(LINUIX)用サンプルソフトメイクファイル
  Mk.cmd           ・・・Windows用サンプルソフトメイクファイル
  apl90.c          ・・・サンプルソフトソースファイル
  sad              ・・・サンプルソフト実行ファイル
```

5-3　ユーザープログラムへの組み込み

DASmini-ModelEシリーズのアプリケーションを作成する場合はインクルードファイル"pci_sad.h"をユーザープログラムへ追加してください。
そしてリンク時に基本サブルーチン"inet_sad_lib.c"をリンクしてください。
UNIX(LINUX)の場合はコンパイルオプションは特にありません。サンプルのMakefileを参照してください。
Windowsの場合はWINNTをdefineしてwsock32.libとuser32.libをリンクしてください。サンプルのmk.cmdを参照してください。

5-4　サンプルソフト使用方法

基本サブルーチンを使用したサンプルプログラムの使用方法を説明します。
ファイル名はdasです。

構文　　das　mode[,parameter_file][,data_file]　[hostname]

modeはDASminiに対する動作の指定です。

INIT	DASminiにイニシャライズコマンドを送ります。
STOP	DASminiにストップコマンドを送ります。
STATus	DASminiのステータスを表示します。
INFO	DASminiの内部情報を表示します。
COND	DASminiの動作状態を表示します。
ADNorm	ADノントリガスタートモードを起動します。
ADTrg	ADトリガスタートモードを起動します。
ADRET	ADリトリガスタートモードを起動します。
ADPRe	ADプリトリガスタートモードを起動します。
ADPOst	ADポストトリガスタートモードを起動します。
ADREPost	ADリポストトリガスタートモードを起動します。

小文字の部分は省略可能です。また、入力は大文字、小文字いずれも対応します。

parameter_fileはサンプリング周波数やデータ量などの情報を含んだファイル
です。ファイルがすでに存在している場合はそのファイルを読み込み、存在
しない場合はパラメータ入力を促しファイルを作成します。
parameter_fileを指定しない場合は"def.par"というファイルが作成されます。

data_fileはADする場合ADデータを格納するファイルになり、DAをする
場合転送するDAデータのファイルになります。
data_fileを指定しなかった場合はADの場合バッファに読み込まれるだけです。
DAの場合12bitのインクリメントパターンを出力します。

HostnameはDASminiのホスト名もしくはIPアドレスを指定します。

パラメータファイルの作成

指定したパラメータがない場合パラメータファイルを作成する為に入力を促してきます。
次に各入力について述べます。

① AD 及び DA 共に出力されます。

[1] Use internal clock
[2] Use external clock
[3] Use prescale external clock
Select no.
計測するクロックソースを内部,外部又はプリスケール（分周）外部クロックにするか指定します。
内部の場合 1,外部の場合 2,プリスケール（分周）外部クロックの場合 3 を入力します。

②-1 内部クロックを指定した場合出力されます。

[1] Hz unit
[2] KHz unit
[3] uSEC unit
Select no.
計測するクロックの周波数をどの単位で入力するか指定します。Hz で入力したい場合 1,KHz で入力したい場合 2,μSEC で入力したい場合 3

Input(Hz/KHz/μsec)rate
と出力されたら計測したい値を入力してリターンキーを押します。

②-2 プリスケール外部クロックを指定した場合、出力されます。

Input prescale counter

プリスケール値（分周値）を入力します。

③ トリガを使用するモードを指定した場合出力されます。
トリガソースを指定します。

[1] Use external trigger
[2] Use channel trigger
Select no.
TRIG IN 端子を使用する場合は１、入力チャンネルを使用する場合は２を入力します。但し、ＡＤモジュールがない場合は必ず１を設定してください。

③-1 ③にて２（チャンネルトリガ）を指定した場合に出力されます。

Select channel no.[1-1024]

トリガに使用するＡＤチャンネル番号を指定します。（指定ＡＤチャンネルは他のチャンネルと同様に計測にも使用できます）

④　トリガモードを指定した場合出力されます。

[1]Use　positive　edge trigger
[2]Use　negative　edge trigger

Select　no.

トリガの立ち上がり,立ち下がりの指定をします。立ち上がりでトリガする場合 1、 立ち下がりでトリガする場合 2 を設定します。

次に、トリガレベルを聞いきますので、０~１２７の値を設定します。
（下記の表を参照）

Set trigger level.[0-127]

設定値	電圧	±５V	±１０V
DEC			
127	+FS(63/64)v	+4.92V	+9.84V
126	+FS(62/64)v	+4.84V	+9.68V
｜	｜	｜	｜
65	+FS(1/64)v	+0.08V	+0.16V
64	0v	0V	0V
63	-FS(1/64)v	-0.08V	-0.16V
｜	｜	｜	｜
1	-FS(63/64)v	-4.92V	-9.84V
0	-FS(64/64)v	-5V	-10V

＊）1LSB=2FS/128　　入力電圧±５Vの時は　0.078125V
入力電圧±１０Vの時は 0.15625V

⑤-1　AD リポストトリガモードを指定した場合出力されます。
Input　posttrg size

トリガから、遅延するクロック数を設定します。サンプリングクロック設定に関連します。

Input re-posttrg count

くり返し回数を設定します。

⑤-2　AD ポストトリガモードを指定した場合出力されます。

Input　posttrg size

トリガから、遅延するクロック数を設定します。サンプリングクロック設定に関連します。

⑤-3　AD プリトリガモードを指定した場合出力されます。

Input　pretrg　size

トリガ以前にほしい１チャンネルに対するデータ量を指定します。

⑤-4　AD リトリガモードを指定した場合出力されます。

Input re-trg count

くり返し回数を設定します。

⑤-５　DA モードを指定した場合出力されます。

Input auto start size

DA を起動する前に転送する１チャンネルに対するデータ量を指定します。

⑥　AD 及び DA 共に出力されます。

Input　frame　size

AD 及び DA する１チャンネルに対するデータ量を指定します。

⑦ AD 及び DA 共に出力されます。

Normal Randum channel NO. ？ Yes=0／NO=1

ランダムチャンネル設定を標準（１チャンネルから順番）で自動設定するか、１チャンネルずる手動に入力するかを選択します。標準の場合は０、手動設定の場合は１

手動設定を選択した場合は、チャンネル数分、１番目から順に設定チャンネル番号を聞いてきます。

Input　＊th channel number

以上の入力を終了すると、計測を開始します。