

DASBOX Model-Eシリーズ

DASmini-E2800シリーズ

MATLABインタフェース仕様書

作成 平成17年 1月

改訂 平成17年 2月

改訂 平成17年 10月

改訂 平成22年 9月

目 次

改訂履歴	4
第1章 概 要	5
第2章関数説明	6
第3章使用方法	9
3.1 inet_io_open	9
3.2 inet_io_close	9
3.3 inet_io_packet	10
3.4 inet_io_adread	10
3.5 inet_io_adfile	11
3.6 inet_io_dawrite	11
3.7 inet_io_dafile	12
3.8 inet_io_dawait	12
3.9 inet_io_adstart	13
3.10 inet_io_dastart	13
3.11 inet_io_daclear	13
3.12 inet_io_init	14
3.13 inet_io_pre	14
3.14 inet_io_blkf	15
3.15 inet_io_cond	16
3.16 inet_io_stat	17
3.17 inet_amp_set	17
3.18 inet_amp_cutoff	18
3.19 inet_amp_bcutoff	19
3.20 inet_io_adda_cal (24bit機専用)	20
3.21 inet_io_clock_sync (24bit機専用)	20
3.22 inet_io_mute (24bit機専用)	21
3.23 inet_io_async_adread	21
3.24 inet_io_async_dawrite	22
3.24 inet_io_async_adread_dawrite	23
3.25 inet_io_wait_adread	24
3.24 inet_io_wait_dawrite	25
第4章アーギュメントの説明	26
4.1 DASBOX アーギュメント	26

4.1.1	DASARG.dasmode.....	27
4.1.2	DASARG.stat.....	27
4.1.3	DASARG.dmatime	28
4.1.4	DASARG.attn	28
4.1.5	DASARG.gain.....	28
4.1.6	DASARG.trgslp	28
4.1.7	DASARG.fifo.....	29
4.1.8	DASARG.clkmode	29
4.1.9	DASARG.clock.....	30
4.1.10	DASARG.frame1	30
4.1.11	DASARG.frame2.....	30
4.1.12	DASARG.frame3	30
4.1.13	DASARG.frame4	30
4.1.14	DASARG.frame5	30
4.1.15	DASARG.trglevel	31
4.1.16	DASARG.trgsrc	31
4.1.17	DASARG.trgch	31
4.1.18	DASARG.channels	32
4.1.19	DASARG.chanun.....	32
4.1.20	DASARG.master	32
4.1.21	DASARG.ext_parm	32
4.1.22	DASARG.dmasize.....	33
4.1.23	DASARG.pre_mem_size	33
4.2	AF アーギュメント	34
4.2.1	AFARG.gain.....	34
4.2.2	AFARG.offval.....	35
4.2.3	AFARG.offmode.....	36
4.2.4	AFARG.input	36
4.2.5	AFARG.imode	36
4.2.7	AFARG.cutoff.....	37
4.3	フィルタ設定アーギュメントの説明	38
4.3.1	BARG.cutoff.....	38

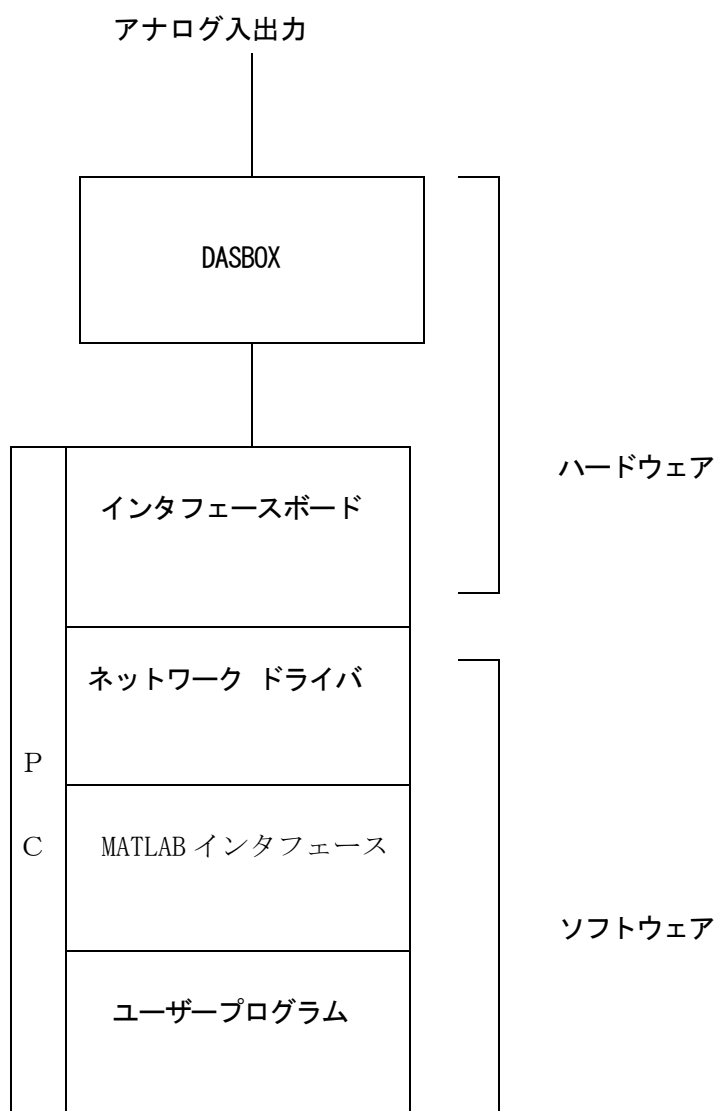
改訂履歴

平成 17 年 2 月 7 日	inet_io_dafile 関数追加
平成 17 年 10 月 30 日	inet_io_open 関数に ipaddress の説明を追加
平成 22 年 9 月 27 日	MATLAB R2009a 対応

第1章 概 要

本 I/F 関数は、**DASBOX Model-E** シリーズ及び **DASmini-E2800** シリーズ
(以降 **DASBOX** と表記) を P C 上で使用するためのプログラムです。

本 I/F 関数は、**DASBOX** 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 DA 出力及びそれらに付随する設定を **MATLAB** 上で行います。



第 2 章関数説明

`inet_io_open()`

DASBOX をオープンします。

`inet_io_close()`

DASBOX をクローズします。

`inet_io_packet()`

DASBOX に対して動作を指示します。

`inet_io_adread()`

DASBOX からの AD データを読み込みます。

`inet_io_adfile()`

DASBOX からの AD データを読み込み、ファイルに書き込みます。

`inet_io_dawrite()`

DA データを DASBOX に転送します。

`inet_io_dafile()`

ファイルからデータを読み込み、DASBOX に出力します。

`inet_io_dawait()`

実際の DA が終了するまで待ち状態にします。

`inet_io_daclear()`

指定されたチャンネルの DA のアナログ出力を 0 V に変化させます。

`inet_io_adstart()`

DASBOX に対して AD スタートコマンドを送信します。

`inet_io_dastart()`

DASBOX に対して DA スタートコマンドを送信します。

`inet_io_init()`

D A S B O X に対してイニシャライズコマンドを送信します。

`inet_io_pre()`

プリトリガ動作を行なった時、トリガ以前の無効データ数を返します。

`inet_io_blkf()`

データ転送のブロックサイズを設定します。

`inet_io_cond()`

D A S B O X の状態を調べます。

`inet_io_stat()`

A D / D A 動作中にエラーが発生した時、D A S B O X のエラーステータスを読み取ります。

`inet_amp_set()`

アンプ関連項目の設定を行います。（オプション）

`inet_amp_cutoff()`

フィルタ関連項目の設定を行います。（オプション）

`inet_amp_bcutoff()`

フィルタ関連項目の設定を行います。（オプション）

`inet_io_adda_cal()`

D A S B O X (24bit 機)内の AD オフセット・キャリブレーションを行います。

`inet_io_clock_sync()`

複数台の D A S B O X (24bit 機)のサンプリングクロックの位相を同期させます。

`inet_io_mute()`

D A S B O X (24bit 機)内の全ての D A 出力に **mute** をかけます。

`inet_io_async_adread()`

D A S B O X からの AD データを非同期に読み込みます。

`inet_io_async_dawrite()`

D A データを D A S B O X に非同期に転送します。

`inet_io_async_dawrite_adread()`

D A データを D A S B O X に非同期に転送し、AD データを非同期に読み込みます。

`inet_io_wait_adread()`

`inet_io_async_adread()`の終了を待ちます。

`inet_io_wait_dawrite()`

`inet_io_async_dawrite()`の終了を待ちます。

第 3 章使用方法

3.1 inet_io_open

機能：DASBOXをオープンします。

デバイスオープン後、DASBOXに対し、インフォメーションコマンドを発行します。以降の関数は、本関数呼出し後動作可能になります。

形式： 戻り値 = inet_io_open(DAS 番号, ipaddress)

DAS 番号には、実数の整数の値を指定して下さい。

範囲： 1～接続しているDASBOXの台数

注：DAS 番号は関数によって複数台指定できるものと 1 台しか指定できないものがあります。以下説明していないものは複数台指定できます。

ipaddress には、DASBOX の IP ADDRESS を指定します。

例 1 rval=inet_io_open(1, '192.168.0.2');

例 2 id = [1 2];

ip = {'192.168.0.2' '192.168.0.3'};

rval=inet_io_open(id, ip);

エラー時は戻り値に NaN を返します。

3.2 inet_io_close

機能：DASBOXをクローズします。

形式： 戻り値 = inet_io_close(DAS 番号)

DAS 番号には、inet_io_open で指定した数値を指定して下さい。

エラー時は戻り値に NaN を返します。

3.3 inet_io_packet

機能：DASBOXに対して動作を指示します。

DASARG. で始まる名前のグローバル変数に登録された内容で、DASBOXに設定します。

形式： 戻り値 = inet_io_packet(DAS 番号 [, BIT16])

DAS 番号には、inet_io_open で指定した数値を指定して下さい。

本関数では複数台を指定できません。必ず 1 台ずつ設定してください。

BIT16 で 16bit/32bit データモードを切り替えます。省略時は 32bit データモードになります。

0 = 32bit データモード

1 = 16bit データモード

エラー時は戻り値に NaN を返します。

3.4 inet_io_adread

機能：DASBOXからのADデータを読み込みます。

形式： ADデータ = inet_io_adread(DAS 番号, ADデータ数)

関数の戻り値として、ADデータ（列ベクトル）を返します。

データは電圧値データです。

複数チャンネルを指定した場合でも同様なので、必要に応じて reshape 関数等で行列の形を変更して下さい。

ADデータ数は、チャンネル×フレーム数です。

本関数では複数台を指定できません。必ず 1 台ずつ設定してください。

エラー時はADデータに NaN がスカラで入ります。

3.5 inet_io_adfile

機能：DASBOXからのADデータを読み込み、ファイルに書き込みます。

形式： 戻り値 = `inet_io_adfile`(DAS 番号, ファイル名, ADデータ数)

ADデータ数は、チャンネル×フレーム数です。

既存のファイルに上書きし、かつ元ファイルが計測データよりサイズが大きい場合、ファイルの先頭からデータ数分のみ上書きされますので注意が必要です。データはバイナリデータで書き込まれます。

本関数では複数台を指定できません。必ず 1 台ずつ設定してください。

エラー時は戻り値に NaN を返します。

3.6 inet_io_dawrite

機能：DAデータをDASBOXに転送します。

形式： 戻り値 = `inet_io_dawrite`(DAS 番号, DAデータ
[,DAデータ数])

DAデータは列ベクトルで指定して下さい。

データは電圧値データです。

複数チャンネルを指定した場合でも同様なので、必要に応じて `reshape` 関数等で行列の形を変更して下さい。

DAデータ数を省略した場合、DAデータを全て転送します。

本関数では複数台を指定できません。必ず 1 台ずつ設定してください。

エラー時は戻り値に NaN を返します。

3.7 inet_io_dafile

機能：ファイルからデータを読み込み、DASBOXに出力します。

形式： 戻り値 = inet_io_dafile(DAS 番号, ファイル名, DA データ数)

DA データ数は、チャンネル×フレーム数です。

エラー時は戻り値に NaN を返します。

DAデータファイルは列ベクトルでセーブしたファイルを使用して下さい。

複数チャンネルを指定した場合でも同様なので、必要に応じて reshape 関数等で行列の形を変更したファイルを作成して下さい。

本関数では複数台を指定できません。必ず 1 台ずつ設定してください。

3.8 inet_io_dawait

機能：実際のDAが終了するまで待ち状態にします。

形式： 戻り値 = inet_io_dawait(DAS 番号, タイムアウト時間)

タイムアウト時間には、以下の定数を指定します。

TNONE	なし
T1s	1 秒
T3s	3 秒
T10s	10 秒
T30s	30 秒
T100s	100 秒
T300s	300 秒
T1000s	1000 秒

戻り値：

0	正常終了
-1	異常終了
-2	タイムアウト

3.9 inet_io_adstart

機能：DASBOXへADスタートコマンドを送信します。

形式： 戻り値 = inet_io_adstart(DAS 番号)

DAS 番号には、inet_io_open で指定した数値を指定して下さい。

エラー時は戻り値に NaN を返します。

3.10 inet_io_dastart

機能：DASBOXへDAスタートコマンドを送信します。

形式： 戻り値 = inet_io_dastart(DAS 番号)

DAS 番号には、inet_io_open で指定した数値を指定して下さい。

エラー時は戻り値に NaN を返します。

3.11 inet_io_daclear

機能：DASBOXのDAのアナログ出力を0Vに変化させます。

形式： 戻り値 = inet_io_daclear(DAS 番号)

DAS 番号には、inet_io_open で指定した数値を指定して下さい。

エラー時は戻り値に NaN を返します。

3.12 inet_io_init

機能：D A S B O Xを初期状態にします。

形式： 戻り値 = inet_io_init(D A S 番号)

D A S 番号には、inet_io_open で指定した数値を指定して下さい。

エラー時は戻り値に NaN を返します。

3.13 inet_io_pre

機能：プリトリガスタートAD実行後、トリガ以前の無効データ数を得る。

形式： 無効データ数 = inet_io_pre(D A S 番号)

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（チャンネル全ての総数）を返します。プリトリガデータがすべて有効であれば、0を返します。

本関数では複数台を指定できません。必ず1台ずつ設定してください。

エラー時は無効データ数に NaN を返します。

3.14 inet_io_blkf

機能： 1 チャンネルに対する1回のinet_io_adread及びinet_io_dawrite
のデータの転送数（ブロック数）を指定します。
inet_io_adstart()及びinet_io_dastart()の前に 1 度だけ指定して下さい。

形式： 戻り値 = inet_blkf(D A S 番号, size)

D A S 番号には、inet_io_open で指定した数値を指定して下さい。
size には、設定する転送数をスカラー値で指定して下さい。
複数台指定できますが size は全て共通の値になります。

エラー時は戻り値に NaN を返します。

3.15 inet_io_cond

機能：DASBOX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(),inet_io_dastart()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： DASBOX の状態 = inet_io_cond(DAS 番号)

本関数では複数台を指定できません。必ず1台ずつ設定してください。

DASBOXの状態：各状態が重なった場合の数値は加算された値（ビットのOR）となります。

DAS_SAMP_ERR(512)

DASBOXが計測中にサンプリングエラー又はFiFoオーバーフローエラーとなった場合に”512”となり。inet_io_adstart(),inet_io_dastart(),inet_io_init()で、“0”クリアされます。

DAS_RUN (128)

DASBOXが計測中に“128”となり、計測終了すると“0”となります。

DAS_TRIG (32)

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“32”となります。

エラー時はDASBOXの状態にNaNを返します。

3.16 inet_io_stat

機能：DASBOX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(),inet_io_dawrite()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：

エラーステータス = inet_io_stat(D A S 番号);

本関数では複数台を指定できません。必ず1台ずつ設定してください。

エラーステータス：

0	正常終了
32768(0x8000)	サンプリングエラー
32774(0x8006)	バッファエラー

エラー時はエラーステータスに NaN を返します。

3.17 inet_amp_set

機能：指定したチャンネルにアンプ関連項目の設定を行います。

DASBOX-E シリーズに対応したアンプフィルタモジュール、DASmini-E2800 の“AF”モデルに対応しています。

AFARG. で始まる名前のグローバル変数に登録された内容で、DASBOXに設定します。

形式： 戻り値 = inet_amp_set (D A S 番号)

D A S 番号には、inet_io_open で指定した数値を指定して下さい。

本関数では複数台を指定できません。必ず1台ずつ設定してください。

エラー時は戻り値に NaN を返します。

3.18 inet_amp_cutoff

機能：全てのチャンネルにフィルタ関連項目の設定を行います。
DASBOX-E シリーズに対応したアンプフィルタモジュール、
DASmini-E2800 の“AF”モデルに対応しています。

AFARG.cutoff に登録されたカットオフ周波数を D A S B O X に設定
します。

形式： 戻り値 = inet_amp_cutoff (D A S 番号)

D A S 番号には、inet_io_open で指定した数値を指定して下さい。
本関数では複数台を指定できません。必ず 1 台ずつ設定してください。

エラー時は戻り値に NaN を返します。

3.19 inet_amp_bcutoff

機能：8 チャンネルのブロック単位にフィルタ関連項目の設定を行います。
DASBOX-E シリーズに対応したアンプフィルタモジュール、
DASmini-E2800 の“AF”モデルに対応しています。

BARG.cutoff に登録されたカットオフ周波数をDASBOXに設定
します。

形式： 戻り値 = inet_amp_bcutoff (DAS 番号 , block)

DAS 番号には、inet_io_open で指定した数値を指定して下さい。
本関数では複数台を指定できません。必ず 1 台ずつ設定してください。
block には、設定するブロックを指定して下さい。

例)

block = 0 ・・・全てのチャンネル (cutoff[0]～[15])
block = 1 ・・・1 チャンネル～8 チャンネル (cutoff[0])
block = 2 ・・・9 チャンネル～16 チャンネル (cutoff[1])

エラー時は戻り値に NaN を返します。

3.20 inet_io_adda_cal (24bit 機専用)

機能：DASBOX(24bit 機)内の AD のオフセット・キャリブレーションを行います。ADC の DC オフセットの自動構成を行う関数です。

必ずしも毎回の計測で行う必要はありませんが、装置の電源投入後 1 回は行うことを推奨します。またこの関数で DASBOX アーギュメントメンバーの clock 設定値を使用しますので、計即時の値を入れておく必要があります。

形式： 戻り値 = inet_io_adda_cal (D A S 番号)

D A S 番号には、inet_io_open で指定した数値を指定して下さい。

エラー時は戻り値に NaN を返します。

3.21 inet_io_clock_sync (24bit 機専用)

機能：DASBOX(24bit 機)に対して動作を指示します。

複数台の DASBOX のサンプリングクロックの位相を同期させます。

形式： 戻り値 = inet_io_clock_sync (D A S 番号)

D A S 番号には、inet_io_open で指定した数値を指定して下さい。

エラー時は戻り値に NaN を返します。

3.22 inet_io_mute (24bit 機専用)

機能：DASBOX(24bit 機)内の全ての DA 出力に mute をかけます。

形式： 戻り値 = inet_io_mute (D A S 番号 , MUTE_ON)

D A S 番号には、inet_io_open で指定した数値を指定して下さい。

MUTE_ON で、mute の ON/OFF を切り替えます。

0 = mute を OFF にします。

1 = mute を ON にします。

エラー時は戻り値に NaN を返します。

3.23 inet_io_async_adread

機能：複数台のDASBOXからADデータを非同期に読み込みます。

形式： ADデータ = inet_io_adread(D A S 番号, ADデータ数)

関数の戻り値として、ADデータ（行列ベクトル[フレーム数,総チャンネル数]）を返します。

データは電圧値データです。

DAS 番号は 1 と指定しても良いですし、[1 2 3]と指定しても構いません。

ADデータ数は、フレーム数×総チャンネルです。

例えば3台のDASBOXでチャンネル数がそれぞれ1ch,2ch,3chと設定されていた場合 1+2+3=6 で総チャンネル数の6の倍数で指定してください。

エラー時はADデータに NaN がスカラで入ります。

3.24 inet_io_async_dawrite

機能：DAデータを複数台のDASBOXに非同期に転送します。

形式： 戻り値 = inet_io_dawrite(DAS 番号, DA データ, DA データ数)

DAデータは行列ベクトル[フレーム数,総チャンネル数]で指定して下さい。

データは電圧値データです。

DA データ数はフレーム数×総チャンネルです。

例えば3台のDASBOXでチャンネル数がそれぞれ1ch,2ch,3chと設定されていた場合 $1+2+3=6$ で総チャンネル数の6の倍数で指定します。

エラー時は戻り値に NaN を返します。

3.24 inet_io_async_adread_dawrite

機能：DAデータを複数台のDASBOXに転送すると同時にADデータを複数台のDASBOXから読み込みます。

形式： 戻り値 = inet_io_async_adread_dawrite(DAS番号, ADデータ数, DAデータ, DAデータ数)

DAデータは行列ベクトル[DAフレーム数,総DAチャンネル数]で指定して下さい。

関数の戻り値として、ADデータ（行列ベクトル[ADフレーム数,総ADチャンネル数]）を返します。

データは電圧値データです。

ADデータ数はADフレーム数×総ADチャンネル、DAデータ数はDAフレーム数×総DAチャンネルです。

例えば3台のDASBOXでADチャンネル数がそれぞれ1ch,2ch,3chと設定されていた場合 $1+2+3=6$ で6の倍数でADデータ数を指定します。DAデータ数も同様です。

エラー時は戻り値に NaN を返します。

3.25 inet_io_wait_adread

機能：inet_io_async_adread()の終了を待ちます。

形式： 戻り値 = inet_io_wait_adread(D A S 番号)

3.24 inet_io_wait_dawrite

機能：inet_io_async_dawrite()の終了を待ちます。

形式： 戻り値 = inet_io_wait_dawrite(D A S 番号)

第4章アーギュメントの説明

4.1 DASBOX アーギュメント

DASBOXに対して動作を指示するグローバル変数で、`inet_io_packet` ファンクションで使います。

グローバル変数の定義は、M-スクリプト `inet_io_define.m` で行ないます。

ユーザーは、本関数を使用する場合、当スクリプトを実行して使います。

DASBOXを動作の動作を設定するには、以下のアーギュメントを格納するグローバル変数を設定した後、`inet_io_packet` 関数を使います。

`DASARG.dasmode`
`DASARG.dasstat`
`DASARG.dmatime`
`DASARG.attn`
`DASARG.gain`
`DASARG.trgslp`
`DASARG.clkmode`
`DASARG.fifo`
`DASARG.clock`
`DASARG.frame1`
`DASARG.frame2`
`DASARG.frame3`
`DASARG.frame4`
`DASARG.frame5`
`DASARG.mutelevel`
`DASARG.trglevel`
`DASARG.trgsrc`
`DASARG.trgch`
`DASARG.channels`
`DASARG.chanum`
`DASARG.master`
`DASARG.ext_parm`
`DASARG.dmasize`

DASARG.pre_mem_size

4.1.1 DASARG.dasmode

DASBOXに対する動作を指定します。

定数	値	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リトリガスタートモード
DANORM	4	DAノントリガスタートモード
DATRG 5		DAノーマルトリガスタートモード
DARET 6		DAリトリガスタートモード
DACYCL	7	DAノントリガスタートサイクルモード
DATCYC	8	DAノーマルトリガスタートサイクルモード
DARCYC	9	DAリトリガスタートサイクルモード

4.1.2 DASARG.stat

未使用

4.1.3 DASARG.dmatime

inet_io_adread, inet_io_dawrite で指定するバッファサイズ分のタイムアウト
値を秒数で設定します。

4.1.4 DASARG.attn

DASBOXにSRQ出力の指示を与えます。

値	動作
---	----

0	下記SRQを発生させない。
---	---------------

4096	DA終了時にSRQを発生させる。(0x1000)
------	--------------------------

4.1.5 DASARG.gain

DASBOXの入力GAINの指示を与えます。

値

0	GAIN1
---	-------

1	GAIN2
---	-------

4.1.6 DASARG.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

0 以外	立ち上がりでトリガ
------	-----------

0	立ち下がりでトリガ
---	-----------

4.1.7 DASARG.fifo

DASBOXのFIFOサイズが `inet_io_open` を行なった時
ワードサイズで設定されます。
ユーザーが設定する必要はありません。

4.1.8 DASARG.clkmode

サンプリングクロックのソースを指定します。

定数	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
PRECLK	3	外部からクロックを分周する

4.1.9 DASARG.clock

内部クロック及び外部分周を指定した場合有効で、内部クロックの場合は周波数をH z 単位で指定し、外部分周の場合は分周値を指定します。

4.1.10 DASARG.frame1

A D又はD Aするデータ量を指定します。

4.1.11 DASARG.frame2

プリトリガ動作を指定した場合有効でトリガ以前のデータ量を指定します。

4.1.12 DASARG.frame3

D A動作を指定した場合有効でD Aが実際に起動する前に転送するデータ量を指定します。

4.1.13 DASARG.frame4

D Aサイクル動作を指定した場合有効で繰り返す回数を指定します。

4.1.14 DASARG.frame5

リトリガモードを指定した場合、トリガで繰り返す回数を指定します。但し、D Aリトリガスタートサイクルモードの場合、トリガで繰り返す回数は、DASARG.frame4 が優先されます。

4.1.15 DASARG.trglevel

DASBOXがトリガレベル機能を持っている時、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。

範囲： 0 ～ 127

値	電圧	
127	+FSv	
64	0v	(1タップ=FS/64)
0	-FSv	

4.1.16 DASARG.trgsrc

各トリガモードを選択した場合有効で、トリガ入力ソースを指定します。

値	トリガ入力ソース
0	AD入力チャンネルトリガ
1	外部トリガ端子

4.1.17 DASARG.trgch

トリガソースをAD入力チャンネルに指定、そのチャンネル番号を指定します。

値	ADチャンネル番号
1 ～ 1024	NO. 1～NO. 1024

4.1.18 DASARG.channels

AD又はDAするチャンネル数を指定します。

範囲 1～1024

4.1.19 DASARG.chanun

AD又はDAするチャンネル番号を行ベクトルで指定します。

例： DASARG.chanum = 1:1024;

4.1.20 DASARG.master

マスターモード、スレーブモードを指定します。

値： Master/Slave

0 スレーブモードとなり、マスターからクロックを供給します。

1 マスターモードとなり、スレーブにクロックを供給します。
1台で使用する場合はマスター設定としてください。

4.1.21 DASARG.ext_parm

DASmini-E2800 の 24 ビット機種の場合有効です。24 ビット機種の DA/AD の同期をとる場合に DA(マスター側)の同期クロックを 1:1 で出力するか 1/2 分周して出力するかの指定を行います。

詳しくは 24 ビット機種の AD/DA 同期説明.pdf を参照してください。

DASBOX-E シリーズではこのメンバーは使用しません。

値： 同期クロック出力

0 1:1(標準)

1 1/2

4.1.22 DASARG.dmasize

未使用

4.1.23 DASARG.pre_mem_size

未使用

4.2 AF アーギュメント

アンプフィルタの機能を指示するグローバル変数で、`inet_amp_set` 及び、`inet_amp_cutoff` ファンクションで使します。

グローバル変数の定義は、M-スクリプト `inet_io_define.m` で行ないます。ユーザーは、本関数を使用する場合、当スクリプトを実行して使します。アンプフィルタの機能を設定するには、以下のアーギュメントを格納するグローバル変数を設定した後、`inet_amp_set` 及び、`inet_amp_cutoff` 関数を使します。

4.2.1 AFARG.gain

ゲインを設定する値を行ベクトルで指定します。

例： `AFARG.gain = ones(1,128);`

設定値	入力レベル	
0	±10V	(0.5 倍)
1	± 5V	(1 倍)
2	±2.5V	(2 倍)
3	±1.25V	(4 倍)
4	±625mV	(8 倍)
5	±312.5mV	(16 倍)
6	±156.25mV	(32 倍)
7	±78.25mV	(64 倍)

4.2.2 AFARG.offval

オフセット値を設定する値を行ベクトルで指定します。

例： `AFARG.offval = zeros(1,128);`

オフセット設定機能は、オプションです。通常は、全チャネル、ゼロを指定して下さい。

指定したチャネルに対してオフセット電圧を指定します。

設定値は `offmode` の値により異なります。

設定値	指定電圧値 (offmode:Couse/Fine)
127(7f)	+FS (+5V / +0.5V)
0	0 (0V)
-128(80)	-FS (-5V / -0.5V)

4.2.3 AFARG.offmode

オフセットのモードを設定する値を行ベクトルで指定します。

例： `AFARG.offmode = ones(1,128);`

設定値	オフセットレベル
0	±5.0V (Couse)
1	±0.5V (Fine)

4.2.4 AFARG.input

入出力モード(AC/DC)を設定する値を行ベクトルで指定します。

例： `AFARG.input = zeros(1,128);`

設定値	入力モード
0	DC.Coupling
1	AC.Coupling

4.2.5 AFARG.imode

入力モード(SIGNAL/GND)を設定する値を行ベクトルで指定します。

例： `AFARG.imode = zeros(1,128);`

設定値	入力モード
0	シグナル
1	GND

4.2.7 AFARG.cutoff

全チャネル共通フィルタのカットオフ周波数を設定します。

設定値	カットオフ周波数	
0	10Hz	
1	20Hz	
2	50Hz	
3	100Hz	
4	200Hz	
5	500Hz	
6	1kHz	
7	2kHz	
8	5kHz	
9	10kHz	
1 0	20kHz	
1 1	25kHz	
1 2	-----	
1 3	-----	
1 4	Through	
1 5	EXT.Mode	(オプション)

4.3 フィルタ設定アーギュメントの説明

4.3.1 BARG.cutoff

カットオフ周波数を設定する値を行ベクトルで指定します。

例： `BARG.cutoff = ones(1,16);`

設定値	カットオフ周波数
0	10Hz
1	20Hz
2	50Hz
3	100Hz
4	200Hz
5	500Hz
6	1kHz
7	2kHz
8	5kHz
9	10kHz
1 0	20kHz
1 1	25kHz
1 2	-----
1 3	-----
1 4	Through
1 5	EXT.Mode (オプション)