

*ADSP*シリーズ

ADSP 324-141

ソフトウェア・ユーザーズ・マニュアル

目次

1. 概要	2
2. 機能一覧	2
3. 供給形態	2
4. 供給ファイルの一覧.....	2
5. 関数の一覧	3
6. 関数詳細	4
ボードの初期化およびライブラリーの初期化.....	4
指定チャンネルの A/D 変換.....	5
複数チャンネルの A/D 変換.....	6
指定チャンネルの A/D 連続変換.....	7
複数チャンネルの A/D 連続変換.....	8
A/D 変換終了確認.....	9
指定チャンネルの D/A 変換.....	10
複数チャンネルの D/A 変換.....	11
指定チャンネルの D/A 連続変換.....	12
複数チャンネルの D/A 連続変換.....	13
D/A 変換終了確認.....	14
A/D + D/A 同期動作.....	15
A/D + D/A 同期変換終了確認.....	16
7. 構造体の説明	17
1) A/D & D/A ボードの定義.....	17
8. ボード制御ソフトを書く上での注意.....	18
1) ベクタの使用	18

1. 概要

ADSP324-141 サポートソフトウェアは、ADSP324-141 を使用するための基本機能を含んだBIOSプログラム（A141BIOS）および、それを用いたサンプルプログラムから構成されています。

A141BIOSはCで記述されており、制御用に使用する上で大きな手掛かりとなると思われます。

2. 機能一覧

A141BIOSには次の機能があります。

ADSP324-141 ボードの初期化
ソフトウェア同期のA/D&D/A変換機能
タイマー同期のA/D&D/A変換機能

3. 供給形態

A141BIOSはソースファイルおよび、COFFファイル形式のオブジェクトで供給されます。ユーザープログラムとリンクして使用してください。

4. 供給ファイルの一覧

README.DOC	A141BIOS の簡単な説明が書かれています。
A141BIOS.C	A141BIOS のソースファイル
A141BIOS.H	A141BIOS のヘッダファイル
A141BIOS.OBJ	A141BIOS のオブジェクトファイル
ADSPTYPE.OBJ	A141BIOS 内にて使用
SAMPL.BAT	SAMPL.C をコンパイルするためのバッチファイル
SAMPL.C	A141BIOS を使用したサンプルソフト
SAMPL.LNK	SAMPL.OUT を作成する為のリンク用ファイル

5. 関数の一覧

○初期化関数

A32X_141init ボードの初期化および、ライブラリーの初期化をおこなう。

○A/D変換関数

A32X_ADsingle 指定チャンネルのA/D変換
A32X_ADmulti 複数チャンネルのA/D変換
A32X_ADrepeats 指定チャンネルのA/D連続変換
A32X_ADrepeatm 複数チャンネルのA/D連続変換
A32X_ADstatus A/D連続変換終了確認

○D/A変換関数

A32X_DAsingle 指定チャンネルのD/A変換
A32X_DAmulti 複数チャンネルのD/A変換
A32X_DArepeats 指定チャンネルのD/A連続変換
A32X_DArepeatm 複数チャンネルのD/A連続変換
A32X_DAstatus D/A連続変換終了確認

○A/D & D/A同時変換

A32X_ADAsync 複数チャンネルのA/D & D/A連続連続変換
A32X_ADAsatus A/D & D/A連続変換終了確認

6. 関数詳細

関数名

ボードの初期化およびライブラリーの初期化

記述

```
int A32X_141init(int max, unsigned long base)
```

引き数

max	ボード実装数
base	最初のベースアドレス

戻り値

-1	初期化失敗(パラメータ異常)
0	初期化正常終了

説明

ADSP324-141を初期化(D/Aの出力を0Vに設定)します。
またライブラリーの諸設定をおこないます。
ボード実装枚数の指定は、1～8が設定可能です。
ボードのベースアドレスは、1枚目のボードから20hステップで連続して設定し、
最初のボードのベースアドレスを与えてください。
初期化構造体の説明は、第7章. 構造体の説明を参照してください。

使用例

```
#include "a141bios.h"

#define BD_MAX 8

A141BD_PORT *port = (A141BD_PORT *)0x900280

void main() {
    A32X_141init(BD_MAX, port);
}
```

関数名

指定チャンネルの A/D 変換

記述

```
int A32X_ADsingle(int ch, float *data);
```

引き数

ch	チャンネル番号(0～[実装枚数*16-1])
*data	A/D 変換結果

戻り値

- 1	初期化失敗(パラメータ異常)
0	変換正常終了

説明

指定されたチャンネルを A/D 変換します。

使用例

```
#include "a141bios.h"

float AD_BUF;

void main() {
    A32X_ADsingle(0, AD_BUF);
}
```

関数名

複数チャンネルの A/D 変換

記述

```
int A32X_ADmulti(int ch, float *data);
```

引き数

ch	チャンネル数
*data	データ格納先ポインタ

戻り値

-1	初期化失敗(パラメータ異常)
0	変換正常終了

説明

複数チャンネルを A/D 変換します。

使用例

```
#include "a141bios.h"

float AD_BUF[16];

void main() {
    A32X_ADmulti(16, AD_BUF);
}
```


関数名

指定チャンネルの A/D 連続変換

記述

```
int A32X_ADrepeats(float prod, int ch, unsigned long size, float *buff);
```

引き数

prod	取り込み周期 (μ SEC) 20 μ SEC 以下は指定しないでください。
ch	チャンネル番号
size	取り込みサイズ
*buff	データ格納ポインタ

戻り値

-1	変換失敗 (パラメータ異常)
0	変換正常終了

説明

指定されたチャンネルを A/D 変換します。
変換データ数は、変換するデータサイズを指定します。
変換周期には、変換間隔を μ 秒単位で指定します。
DSW104 は最初のボードからそれぞれ 1, 2, 3... 8 の各位置を ON にしてください。

使用例

```
#include "a141bios.h"

#define AD_BUF[32];

void main() {
    A32X_ADrepeats(100, 0, 32, AD_BUF);
}
```

関数名

複数チャンネルの A/D 連続変換

記述

```
int A32X_ADrepeatm(float prod, int ch, unsigned long size, float *buff);
```

引き数

prod	取り込み周期 (μ SEC) 20μ SEC 以下は指定しないでください。
ch	チャンネル数 (16, 32, 48...128)
size	取り込みサイズ
*buff	データ格納ポインタ

戻り値

-1	変換失敗 (パラメータ異常)
0	変換正常終了

説明

複数チャンネルを A/D 変換します。
変換データ数は、変換するデータサイズを指定します。
変換周期には、変換間隔を μ 秒単位で指定します。
データの配列を下記に示します。

buff + 0	先頭チャンネルデータ
buff + 15	最終チャンネルのデータ
buff + 16	次のデータ

以下省略

DSW104 は最初のボードからそれぞれ 1, 2, 3... 8 の各位置を ON にください。

使用例

```
#include "a141bios.h"

float AD_BUF[16*32];

void main() {
    A32X_ADrepeatm(100, 16, 32, AD_BUF);
}
```

関数名

A/D変換終了確認

記述

```
int    A32X_ADstatus(void);
```

引き数

なし

戻り値

1	変換中
0	変換終了

説明

A/D変換の終了を確認します。
この関数は、A32X_ADrepeats, A32X_ADrepeatm とともに使用します。

使用例

```
#include    "a141bios.h"
float      AD_BUF[32]

void    main() {
    A32x_ADrepeats(100, 0, 32, AD_BUF)
    while(A32X_ADstatus());
}
```

関数名

指定チャンネルの D/A 変換

記述

```
int A32X_DAsingle(int ch, float data);
```

引き数

ch	チャンネル番号(0～[実装枚数*12-1])
data	D/A 変換データ

戻り値

- 1	変換失敗(パラメータ異常)
0	変換正常終了

説明

指定チャンネルの D/A 変換を行います。

使用例

```
#include "a141bios.h"

float DA_BUF;

void main() {
    A32X_DAsingle(0, DA_BUF);
}
```

関数名

複数チャンネルの D/A 変換

記述

```
int A32X_DAmulti(int ch, float *data);
```

引き数

ch	チャンネル数
*data	データポインタ

戻り値

-1	変換失敗 (パラメータ異常)
0	変換正常終了

説明

複数チャンネルの D/A 変換を行います。

使用例

```
#include "a141bios.h"

float DA_BUF[12];

void main() {
    A32X_DAmulti(12, DA_BUF);
}
```

関数名

指定チャンネルの D/A 連続変換

記述

```
int A32X_DArepeats(float prod, int ch, unsigned long size, float *buff);
```

引き数

prod	取り込み周期 (μ SEC)
ch	チャンネル番号
size	取り込みサイズ
*buff	データ格納ポインタ

戻り値

- 1	変換失敗 (パラメータ異常)
0	変換正常終了

説明

指定チャンネルの D/A 変換を行います。

変換データ数は、変換するデータサイズを指定します。

変換周期には、変換間隔を μ 秒単位で指定します。

DSW104 は最初のボードからそれぞれ 1, 2, 3... 8 の各位置を ON にしてください。

取り込み周期が早すぎる場合は、取りこぼしが発生する場合があります。

使用例

```
#include "a141bios.h"

float DA_BUF[32];

void main() {
    A32X_DArepeats(100, 0, 32, DA_BUF);
}
```

関数名

複数チャンネルの D/A 連続変換

記述

```
int A32X_DArepeatm(float prod, int ch, unsigned long size, float *buff);
```

引き数

prod	取り込み周期 (μ SEC)
ch	チャンネル数 (12, 24, 36...96)
size	取り込みサイズ
*buff	データ格納ポインタ

戻り値

-1	変換失敗 (パラメータ異常)
0	変換正常終了

説明

複数チャンネルを D/A 変換します。
変換データ数は、変換するデータサイズを指定します。
変換周期には、変換間隔を μ 秒単位で指定します。
データの配列を下記に示します。

buff + 0	先頭チャンネルデータ
buff + 11	最終チャンネルのデータ
buff + 12	次のデータ

以下省略

DSW104 は最初のボードからそれぞれ 1, 2, 3...8 の各位置を ON にください。
取り込み周期が早すぎる場合は、取りこぼしが発生する場合があります。

使用例

```
#include "a141bios.h"

float DA_BUF[12*32];

void main() {
    A32X_DArepeatm(100, 12, 32, DA_BUF);
}
```

関数名

D/A変換終了確認

記述

```
int    A32X_DAstatus(void);
```

引き数

なし

戻り値

1	変換中
0	変換終了

説明

D/A変換の終了を確認します。

この関数は、A32X_DArepeats, A32X_DArepeatm とともに使用します。

使用例

```
#include    "a141bios.h"
float      DA_BUF[32];

void    main() {
    A32X_DArepeats(100, 0, 32, DA_BUF);
    while(A32X_DAstat());
}
```


関数名

A/D + D/A同期動作

記述

```
int A32X_ADAsync(float prod,
                 unsigned long ad_size, float *ad_buff, int ad_chnl,
                 unsigned long da_size, float *da_buff, int da_chnl);
```

引き数

prod	同期周期 (μ SEC)
ad_size	A/D変換データサイズ
ad_buff	A/D変換データ格納ポインタ
ad_chnl	A/Dチャンネル数(16, 32, 48...128)
da_size	D/A変換データサイズ
da_buff	D/A変換データ格納ポインタ
da_chnl	D/Aチャンネル数(12, 24, 36...96)

戻り値

- 1	変換失敗 (パラメータ異常)
0	変換正常終了

説明

複数チャンネルをA/D&D/A変換します。
変換データ数は、変換するデータサイズを指定します。
変換周期には、変換間隔を μ 秒単位で指定します。
データの配列を下記に示します。

ad_buff + 0	A/D先頭チャンネルデータ
ad_buff + 15	A/D最終チャンネルのデータ
ad_buff + 16	A/D次のデータ
以下省略	
da_buff + 0	D/A先頭チャンネルデータ
da_buff + 11	D/A最終チャンネルのデータ
da_buff + 12	D/A次のデータ

以下省略

DSW104は最初のボードからそれぞれ1, 2, 3...8の各位置をONにしてください。
同期周期が早すぎる場合は、同期誤差が発生する場合があります。

```
#include "a141bios.h"

float AD_BUF[16*32];
float DA_BUF[12*32];

void main() {
    A32X_ADAsync(100, 32, AD_BUF, 16, 32, DA_BUF, 12);
}
```

関数名

A/D + D/A同期変換終了確認

記述

```
int A32X_ADAsync(void);
```

引き数

なし

戻り値

1	変換中
0	変換終了

説明

A/D & D/A変換の終了を確認します。
この関数は、A32X_ADAsync とともに使用します。

使用例

```
#include "a141bios.h"
float AD_BUF[16*32];
float DA_BUF[12*32];

void main() {
    A32X_ADAsync(100, 32, AD_BUF, 16, 32, DA_BUF, 12);
    while(A32X_ADAsync(void));
}
```

7. 構造体の説明

構造体定義は typedef を用いて、<a141bios.h>の中で定義されています。

1) A/D&D/Aボードの定義

```
typedef struct{
    unsigned long AD[AD_CH_MAX],           //ADボード配列構造体
                DA[DA_CH_MAX],           //ADチャンネル
                AD_BUSY,                 //DAチャンネル
                AD_CTRL,                 //ADビジーフラグ
                EMPTY,                   //AD制御レジスタ
                INT_RST;                  //未使用
}      A141BD_PORT;                       //割込み要因リセット
```

8. ボード制御ソフトを書く上での注意

ボード制御ソフトをユーザーサイドで独自に作る場合における注意点を説明します。

1) ベクタの使用

割り込みを複数のボード上で使用する上で、ベクタ番号は重要な役割を持ちます。
ベクタ番号の設定は、DSW104で行うことができボード間で重複しないように設定します。

例)

1枚目のボード DSW104-1 をON、他はOFF

2枚目のボード DSW104-2 をON、他はOFF

このように設定しておくことにより、どのボードから割り込み要求がきたか知ることができるようになります。

知る方法は、ベースアドレスの下位16ビット全てが1のアドレス(nnnfffH)番地を読むことによつて行います。

()内のnnは、ボードのベースアドレスの上位8ビットの設定です。

このことから解るとおり、割り込み要求を出しているボードのDSW104のON位置のビットが0になります。

- ・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

ADSP324-141

ソフトウェア・ユーザース・マニュアル

2000.6 第1版発行