

@CROシリーズ

ACRO741—01

ソフトウェア・ユーザーズ・マニュアル

16Bit A/D & D/A ボード

《 目 次 》

1	概説	5
2	機能一覧	5
3	供給形態	5
4	関数一覧	6
5	関数詳細	7
5.1	AP74X_01Init	7
5.2	AP74X_01AiNorm	8
5.3	AP74X_01AoNorm	9
5.4	AP74X_01Ai	10
5.5	AP74X_01AiEndInt	11
5.6	AP74X_01AiExtClk	12
5.7	AP74X_01AiTmr	13
5.8	AP74X_01Ao	14
5.9	AP74X_01AoTmr	15
5.10	AP74X_01TrgWait	16
6	構造体の説明	17

1 概説

ACR0741-01 サポートソフトウェアは、ACR0741-01 を使用するための基本機能を含んだ BIOS ライブラリ (AP01BIOS.a) 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。

2 機能一覧

AP01BIOS には以下の機能があります。

- ・ ACR0741-01 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
- ・ トリガー待機機能

3 供給形態

AP01BIOS はソースファイル及び、ライブラリ形式で供給されています。
オブジェクトリンク又はライブラリリンクのいずれかの方法で利用してください。

4 関数一覧

- 初期化関数
AP74X_01Init ボードの初期化及びライブラリの初期化を行います

- 正規化関数
AP74X_01AiNorm A/D データを浮動小数点形式に変換します
AP74X_01AoNorm 浮動小数点形式データを D/A データに変換します

- A/D 変換関数
AP74X_01Ai 指定 A/D チャンネルの A/D 変換を行います
AP74X_01AiEndInt 指定 A/D チャンネルの A/D 変換を行います
 (A/D 変換完了割込み)
AP74X_01AiExtClk 指定 A/D チャンネルの A/D 変換を行います
 (外部クロック同期)
AP74X_01AiTmr 指定 A/D チャンネルの A/D 変換を行います
 (タイマー同期)

- D/A 変換関数
AP74X_01Ao 指定 D/A チャンネルの D/A 変換を行います
AP74X_01AoTmr 指定 D/A チャンネルの D/A 変換を行います
 (タイマー同期)

- トリガー関数
AP74X_01TrgWait トリガー入力を待ちます

5 関数詳細

5.1 AP74X_01Init

【記述】 `int AP74X_01Init(*adrs, *param)`

【引数】 `AP01_PORT *adrs` ベースアドレス
`AP01_PARAM *param` 初期化パラメータ

【戻り値】 0:正常終了 -1:異常終了

【説明】 ACR0741-01 の初期化（各 CH 毎の変換レンジの設定および D/A 出力を 0[V] に設定）します。
また、ライブラリの諸設定を行います。
初期化構造体の詳細は、第 6 章を参照して下さい。

【使用例】

```
#define BD_BASE 0x82408000
AP01_PARAM Param = {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0}};
AP01_PORT *Port = (AP01_PORT *)BD_BASE;

if( AP74X_01Init(Port, &Param) ){
    printf("初期化エラー\n");
}else{
    printf("初期化完了\n");
}
```

5.2 AP74X_01AiNorm

【記述】 `int AP74X_01AiNorm(top, last, dtc, *src, *dst)`

【引数】

<code>int top</code>	先頭チャンネル番号
<code>int last</code>	最終チャンネル番号
<code>int dtc</code>	変換データ数
<code>int *src</code>	変換元データの格納ポインタ
<code>float *dst</code>	正規化データの格納ポインタ

【戻り値】 0:正常終了 -1:異常終了

【説明】 A/D 変換されたデータを、設定された A/D の入力レンジを基に浮動小数点形式の値に変換します。入力レンジは初期化時の値が使用されます。

【使用例】

```
int i;
int buf[6];
float data[6];

AP74X_01Ai(0, 5, buf);
AP74X_01AiNorm(0, 5, 1, buf, data);
For( i=0; i<6; i++ ){
    printf("CH%d:data=%7.4fYn", i, data[i]);
}
```


5.3 AP74X_01AoNorm

【記述】 int AP74X_01AoNorm(top, last, dtc, *src, dst)

【引数】

int	top	先頭チャンネル番号
int	last	最終チャンネル番号
int	dtc	変換データ数
int	*src	変換元データの格納ポインタ
float	*dst	正規化データの格納ポインタ

【戻り値】 0:正常終了 -1:異常終了

【説明】 D/A 変換する浮動小数点形式のデータを、設定された D/A の出力レンジを基に D/A 変換器の出力形式に変換します。出力レンジは初期化時の値が使用されます。

【使用例】

```
int    buf[4];
float  data[4];

data[0] = 1.2;
AP74X_01AoNorm(0, 0, 1, data, buf);
AP74X_01Ao(0, 0, buf);
```

5.4 AP74X_01Ai

【記述】 int AP74X_01Ai(top, last, *buf)

【引数】 int top 先頭チャンネル番号
int last 最終チャンネル番号
int *buf 変換後データの格納ポインタ

【戻り値】 0:正常終了 -1:異常終了

【説明】 指定された範囲のチャンネルに対して、A/D変換・データ入力を実行します。

【使用例】 int i;
int buf[6];
float data[6];

AP74X_01Ai(0, 5, buf);
AP74X_01AiNorm(0, 5, 1, buf, data);
For(i=0; i<6; i++){
 printf("CH%d:data=%7.4f¥n", i, data[i]);
}

5.5 AP74X_01AiEndInt

【記述】 int AP74X_01AiEndInt (top, last, *buf)

【引数】 int top 先頭チャンネル番号
int last 最終チャンネル番号
int *buf 変換後データの格納ポインタ

【戻り値】 0:正常終了 -1:異常終了

【説明】 A/D 変換完了割込みの機能を使用して、A/D 変換を行います。

【使用例】 int i;
int buf[6];
float data[6];

```
AP74X_01AiEndInt(0, 5, buf);  
AP74X_01AoNorm(0, 5, 1, buf, data);  
For( i=0; i<6; i++ ){  
    printf("CH%d:data=%7.4f¥n", i, data[i]);  
}
```

5.6 AP74X_01AiExtClk

【記述】 `int AP74X_01AiExtClk(top, last, *buf)`

【引数】 `int top` 先頭チャンネル番号
`int last` 最終チャンネル番号
`int *buf` 変換後データの格納ポインタ

【戻り値】 0:正常終了 -1:異常終了

【説明】 外部クロックに同期させて、A/D 変換を行います。

【使用例】 `int i;`
`int buf[6];`
`float data[6];`

`AP74X_01AiExtClk(0, 5, buf);`
`AP74X_01AoNorm(0, 5, 1, buf, data);`
`For(i=0; i<6; i++){`
 `printf("CH%d:data=%7.4f¥n", i, data[i]);`
`}`

5.7 AP74X_01AiTmr

【記述】 int AP74X_01AiTmr(top, last, dtc, prod, buf)

【引数】

int	top	先頭チャンネル番号
int	last	最終チャンネル番号
int	dtc	変換データ数
int	prod	変換周期
int	*buf	変換後データの格納ポインタ

【戻り値】 0:正常終了 -1:異常終了

【説明】 タイマーに同期して、A/D 変換を行います。
変換周期の設定値の単位は[usec]です。

【使用例】

```
int i;
int buf[6];
float data[6];
```

```
AP74X_01AiTmr(0, 5, 1, 100, buf);
AP74X_01AiNorm(0, 5, 1, buf, data);
For( i=0; i<6; i++ ){
    printf("CH%d:data=%7.4f¥n", i, data[i]);
}
```

5.8 AP74X_01Ao

【記述】 int AP74X_01Ao(top, last, buf)

【引数】 int top 先頭チャンネル番号
int last 最終チャンネル番号
int *buf 変換後データの格納ポインタ

【戻り値】 0:正常終了 -1:異常終了

【説明】 指定されたチャンネル範囲に D/A 変換を実行します。

【使用例】 int buf[4];
float data[4];

data[0] = 1.2;
AP74X_01AoNorm(0, 0, 1, data, buf);
AP74X_01Ao(0, 0, buf);

5.9 AP74X_01AoTmr

【記述】 int AP74X_01AoTmr(top, last, dtc, prod, *buf)

【引数】

int	top	先頭チャンネル番号
int	last	最終チャンネル番号
int	dtc	変換データ数
int	prod	変換周期
int	*buf	変換後データの格納ポインタ

【戻り値】 0:正常終了 -1:異常終了

【説明】 タイマーに同期して、D/A 変換を行います。
変換周期の設定値の単位は[usec]です。

【使用例】

```
int i;
int buf[4];
float data[4];

for(i=0; ; i=(i+10)%360){
    data[i/10]=(float)(sin(PI*2.0*(float)i/360.0))*1.0;
    j++;
    if(j>35) break;
}
AP74X_01AoNorm(0, 0, 36, data, buf);
AP74X_01AoTmr(0, 0, 36, 100, buf);
```


6 構造体の説明

構造体は『typedef』を用いて“AP01bios.h”内で定義されています。

初期化構造体

```
typedef struct{
    int  AD_RANGE[AD_CH_MAX],      A/D 入力レンジ
        DA_RANGE[DA_CH_MAX];      D/A 出力レンジ
}AP01_PARAM;
```

ポート

```
typedef struct{
    int  ADDTI[AD_CH_MAX],          A/D 入力(整数)
        ADDTF[AD_CH_MAX],          A/D 入力(単精度浮動小数)
        DADTI[DA_CH_MAX],          D/A 出力(整数)
        DADTF[DA_CH_MAX],          D/A 出力(単精度浮動小数)
        BSY,                          ビジーフラグ読出し
        CTRL,                          割込みコントロールレジスタ設定
        TRG_LVL,                       トリガーレベル設定
        INT_RST,                       割込みフラグリセット
        SMODE,                         サンプルモードレジスタ設定
        CMODE;                          コンバータモードレジスタ
}AP01_PORT;
```



【 改 訂 履 歴 】

改訂番号	改訂日付	改訂内容
初版	2010.05.12	初版

- ・ 本製品及び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。
- ・ 本製品及び本マニュアルの内容は著作権法により保護されています。許可なく複製する事はできません。

ACRO741-01
ソフトウェア取扱説明書

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

2010.05 第1版発行