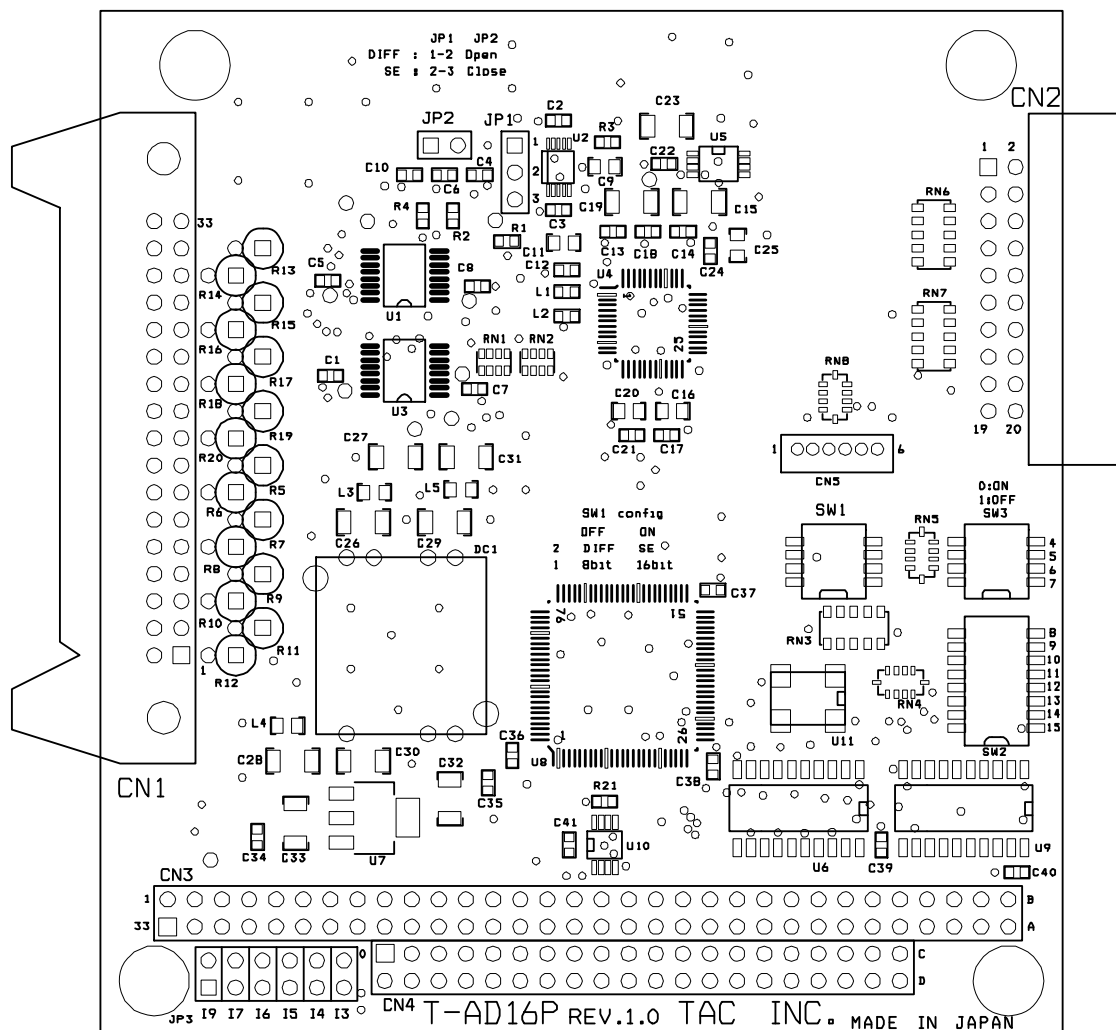


# T-AD16P rev. 1.0

## 取扱説明書

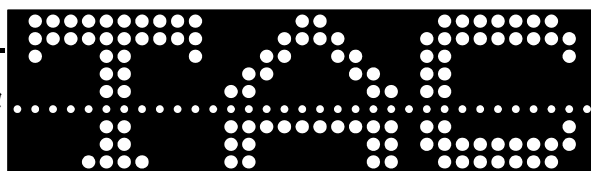
### PC/104バスシリーズ 16ビットADコンバータ 差動8ch・シングルエンド16ch



●各商品は、各社の商標・登録商標です。

●この製品の外観及び仕様は品質改善のため、予告無く変更することがありますのでご了承下さい。

(株)ティーエーシー  
各種制御用マイクロコンピュータ  
設計・製作・販売



〒600-8896  
京都市下京区西七条西石ヶ坪町66  
電話:075-311-7307 FAX:075-314-1174  
<http://www.tacinc.jp>

## はじめに

このたびは、弊社 T-AD16P をお買い上げ頂きましてまことにありがとうございます。このマニュアルは T-AD16P の概要等について説明しています。各 LSI についての詳細は必要に応じてデータシートを参照してください。ハードウェアの不具合に関しましてのサポートはいたしますが、RTOS を含めたソフト面のサポートは基本的にはしておりません。

### 【注意事項】

- (1) 本書の内容に関しては将来予告なしに変更することがあります。
- (2) 本書の一部または全部を無断で転載することは禁止されています。
- (3) 本書の内容については万全を期して作成いたしました。万が一不審な点や誤り、記載もれなどお気づきの点がございましたら、お買い求めの販売店、または当社技術部にご遠慮なくお申しつけください。

### 【本ボードご使用上の注意事項】

- (1) 本ボードは、静電気および衝撃などに十分注意して慎重にお取扱いください。
- (2) 外部入出力電圧、電流は、定格値を越えないよう注意してください。
- (3) コネクタの向き、ピン番号の誤りに注意してください。
- (4) 本ボードの改造及び、その使用にともなった弊害につきましては、当社は一切の責任を負いかねます。

## 1. 仕様概要

PC104 バス 非絶縁 16 ビット AD 変換ボード

アナログデバイセズ AD7612 外部基準電圧 ADR435 使用

8 ビットバス、16 ビットバス対応

差動 8ch または シングルエンド 16ch を設定可能

入力範囲 0 - 5V, 0 - 10V, ±5V, ±10V ソフトウェアで設定

プログラマブルゲインアンプ付き 1, 2, 5, 10 倍 ソフトで設定

スループット 600kSPS (CPU の処理能力に依存します)

変換開始トリガはソフトウェアと外部入力

16 点 汎用 I/O 付き 8 ビット毎に入力、出力を設定可能 外部トリガを使用する場合は IO は 15 点

出力は 3.3V レベル、入力は 5V トレラント、プルアップ済み

## 2. IO アドレスの設定

ベース I/O アドレス(以下 BIO)は SW2 と SW3 で設定します。このスイッチは ON で 0, OFF で 1 です。

出荷時は SW2 の 7 番(A9)と 8 番(A8)のみ OFF にしていますので、BIO は 300H になります。

アドレス	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4
SW2 番号	1	2	3	4	5	6	7	8				
SW3 番号									1	2	3	4
スイッチ状態	ON	ON	ON	ON	ON	ON	OFF	OFF	ON	ON	ON	ON

出荷時の状態

## 3. 差動、シングルエンドの設定

差動、シングルエンドの設定は JP1, JP2, SW1 の 2 番で設定します。

切替は電源が OFF の時に行ってください。

	JP1	JP2	SW1 の 2 番
8CH 差動	1 - 2	オープン	OFF
16CH シングルエンド	2 - 3	ショート	ON

出荷時は 8CH 差動です。

## 4. レジスタ

アドレス	読み込み	書き込み
BIO + 00H	変換データ 下位 8 ビット	AD 変換開始
BIO + 02H	変換データ 上位 8 ビット	
BIO + 04H		入力レンジ CH 選択
BIO + 06H	変換中 ステータス ビット 7 が H の時変換中	ゲイン設定
BIO + 08H	汎用 I/O XPA	汎用 I/O XPA
BIO + 0AH	汎用 I/O XPB	汎用 I/O XPB
BIO + 0CH	汎用 I/O 入出力・外部トリガ設定	汎用 I/O 入出力・外部トリガ設定
BIO + 0EH	ビット 7 : 変換終了割り込みステータス 変換終了で 1	変換終了割り込みクリア 変換終了割り込みステータス

## 5. AD 変換 外部トリガ

AD 変換開始トリガはソフトウェアと外部入力の 2 通りあります。

BIO + 0CH のビット 3 に 1 を書き込むと外部入力トリガが有効になります。

外部入力トリガは CN2 の XPB7(18 番ピン)を使用し、立ち下がりで変換が開始されます。

## 6. AD 変換 手順

- ① BIO + 00H に 0 を書き込むか CN2 の XPB7(18 番ピン)を L で変換開始
- ② BIO + 06H を読み込み ビット 7 が H の時は変換中。
- ③ BIO + 06H のビット 7 が L を確認後、下位 8 ビット (BIO+00H)、上位 8 ビット (BIO+02H)を読み込む  
16 ビットバス対応の CPU モジュールでは BIO+00H を 16 ビット幅で一回で読み込むことが可能

## 7. AD 変換終了割り込み

AD 終了割り込みはラッチを経由し、JP3 へ接続されています。

変換開始トリガはソフトウェアと外部入力の 2 通りあります。

- ① AD 変換終了割り込み発生
- ② 割り込みハンドラから AD 変換値を読み込む。
- ③ BIO + 0EH に 0 を書き込み、割り込みラッチクリア

の手順になります。

### JP3 割り込み 設定

JP3 位置	PC104 バス 割り込み名称
I9	IRQ 9
I7	IRQ 7
I6	IRQ 6
I5	IRQ 5
I4	IRQ 4
I3	IRQ 3

## 8. 入力レンジ CH 選択

入力レンジ、CH 選択、の設定は BIO + 04H にデータを書き込み設定します。

BIO + 04H の bit6 : 0 の時は 2 の補数、1 の時はストレートバイナリ

BIO + 04H の bit5 , bit4 は入力レンジの切替

bit5	bit4	入力レンジ
0	0	0 - 5V
0	1	0 - 10V
1	0	±5V
1	1	±10V

BIO + 04H の bit3 - bit0 は CH 設定です。

CH の設定と CN1 の対応 CN1 は HIROSE HIF3BA-34PA-2.54DS (71) または互換品

Bit3	Bit2	Bit1	Bit0	CH 差動時 コネクタピン番号			CH シングルエンド時 コネクタピン番号	
				+	-	GND	信号	GND
0	0	0	0	1	3	2, 4	1	2
0	0	0	1	5	7	6, 8	5	4
0	0	1	0	9	11	10, 12	9	6
0	0	1	1	13	15	14, 16	13	8
0	1	0	0	17	19	18, 20	17	10
0	1	0	1	21	23	22, 24	21	12
0	1	1	0	25	27	26, 28	25	14
0	1	1	1	29	31	30, 32	29	16
1	0	0	0				3	18
1	0	0	1				7	20
1	0	1	0				11	22
1	0	1	1				15	24
1	1	0	0				19	26
1	1	0	1				23	28
1	1	1	0				27	30
1	1	1	1				31	32

CN1 の 33 番ピンは DCDC コンバータからの +1.5V

CN1 の 34 番ピンは DCDC コンバータからの -1.5V

が出ています。

## 9. プログラマブルゲインアンプの設定

プログラマブルゲインアンプを搭載しております。  
1, 2, 5, 10倍の設定が可能です。

BIO + 06H の bit1, bit0 で設定します。

Bit1	Bit0	ゲイン
0	0	1 倍
0	1	2 倍
1	0	5 倍
1	1	10 倍

## 10. 汎用 I/O

16 点の汎用 I/O を搭載しています。8ビット単位で入力、出力を切り替えることが可能です。  
リセット時は入力ポートです。出力ポートに設定した場合、出力の状態を読むことが可能です。  
出力は標準仕様ではトーテムポールです。標準仕様で出力に設定した場合、1を書き込むと H(3.3V)、0 を書き込むと L です。

アドレスは (BIO + 08H) - (BIO + 0CH) です。

XPA は (BIO + 08H)

XPB は (BIO + 0AH) です。

(BIO + 0CH) のビット0(最下位ビット)XPA の入力/出力の設定ビットです。

(BIO + 0CH) のビット1は XPB の入力/出力の設定ビットです。

1を設定すると出力になります。例えば、(BIO + 0CH) に 3を書き込むと XPA,XPB 共に出力ポートになります。

外部トリガを使用する場合は XPB7 は汎用 I/O として使用できません。

XPA,XPB 共に 10kΩ で 3.3V にプルアップ済みです。

CN2 ピンアサイン 使用コネクタ ヒロセ電機 HIF3FC-20PA-2.54DS または相当品

ピン番号	信号名	ピン番号	信号名
1	GND	2	GND
3	XPA0	4	XPA1
5	XPA2	6	XPA3
7	XPA4	8	XPA5
9	XPA6	10	XPA7
11	XPB0	12	XPB1
13	XPB2	14	XPB3
15	XPB4	16	XPB5
17	XPB6	18	XPB7・外部トリガ
19	3.3V	20	5V

## 11. サンプルプログラム

OS : MES 2.6

CPU : T-SH2MB

ソフトウェアトリガ、外部トリガ、割り込み等

```

#include <mes2.h>
#include <h8/reg704x.h>

//T-AD16P BASE 0x300
#define ADL    (*(volatile unsigned char *)0xc00300)
#define ADU    (*(volatile unsigned char *)0xc00302)
#define ADCH   (*(volatile unsigned char *)0xc00304)
#define ADSTS  (*(volatile unsigned char *)0xc00306)

#define XPA    (*(volatile unsigned char *)0xc00308)
#define XPB    (*(volatile unsigned char *)0xc0030a)
#define XPW    (*(volatile unsigned char *)0xc0030c)

#define IRQC   (*(volatile unsigned char *)0xc0030e)

volatile int it_cnt = 0;
int up_date = 0;
int ad_data;

void int_irq2(void) __attribute__((interrupt_handler));
void int_irq2(void) {
    it_cnt++;
    ad_data = ((ADU << 8) & 0xff00) | (ADL & 0xff);
    up_date = 1;

    ISR &= ~0x20; //clr IRQ2F

    //IRQC = 0; //T-AD16P IRQ_OUT:L
}
void init_irq2(void){
    //PB4 /IRQ2
    PBCR2 &= ~0x0300;
    PBCR2 |= 0x0100;

    set_handler(66, int_irq2);

    ICR1 |= 0x20; //edge
    ICR2 &= ~0x0c00; //down

    ISR &= ~0x20; //clr IRQ2F

    /* pri = 1 */
    IPRA &= ~0x00f0;
    IPRA |= 0x0010;
}

int addata(int a) //0 - 15
{
    a |= 0x40; //straight binary
    a |= 0x10; //0 - 10V or -10V - +10V
    //a |= 0x20; //BI

    ADCH = a; /* select ch. */
    ADCH = a; /* select ch. */
    ADCH = a; /* select ch. */
    ADCH = a; /* select ch. */
    ADCH = a; /* select ch. */
    ADCH = a; /* select ch. */
    ADL = 0; //Start ADconvert
    while((ADSTS & 0x80)){ ; } /* converting */
    return( ((ADU << 8) & 0xff00) | (ADL & 0xff) );
}

```

```

void adin_se(void)
{
    int ch;
    unsigned char c;

    //set gain
    ADSTS = 0; //1
// ADSTS = 1; //2
// ADSTS = 2; //5
// ADSTS = 3; //10

    while( !read(0, &c, 1) ){

        for(ch = 0 ; ch < 16 ; ch ++ ){
            //for(ch = 0 ; ch < 8 ; ch ++ ){
                //if(ch == 8 ) printf(" ");
                printf("%04x ",addata(ch));
                //printf("%05d,",addata(ch));
                //printf("%05d,", addata(ch) >> 1);
            }
            printf("¥r");

            sleep(20);

        }
    }
}

void adin_dif(void)
{
    int ch;
    unsigned char c;

    //set gain
    ADSTS = 0; //1
// ADSTS = 1; //2
// ADSTS = 2; //5
// ADSTS = 3; //10

    while( !read(0, &c, 1) ){

        //for(ch = 0 ; ch < 16 ; ch ++ ){
        for(ch = 0 ; ch < 8 ; ch ++ ){
            if(ch == 8 ) printf(" ");
            printf("%04x ",addata(ch));

                //printf("%05d,",addata(ch));
                //printf("%05d,", addata(ch) >> 1);
        }
        printf(",%04x",XPA);
        printf(",%04x",XPB);
        printf("¥r");

        sleep(20);

    }
}

void adin_irq(void)
{
    int ch;
    unsigned char c;
    int a = 0;
    int data;

```

```

// XPA |= 0x01; //XPA0:H
// XPB |= 0x80; //XPB7:H
XPW = 0x8; //1000 CPLD A IN B IN Enable EXT TRG(PB7)
// XPW = 0x9; //1001 CPLD A OUT B IN Enable EXT TRG(PB7)
// XPW = 0xa; //1010 CPLD A IN B OUT Enable EXT TRG(PB7)

IRQC = 0; //

//set gain
ADSTS = 0; //1
// ADSTS = 1; //2
// ADSTS = 2; //5
// ADSTS = 3; //10
a |= 0x40; //straight binary
a |= 0x10; //0 - 10V or -10V - +10V
//a |= 0x20; //BI

ADCH = a; /* select ch. */

while( !read(0, &c, 1) ){ //hit key ?
//XPB &= ~0x80; //XPB7:L
//XPB |= 0x80; //XPB7:H
//XPA &= ~0x01; //XPA0:L
//XPA |= 0x01; //XPA0:H

if(up_date){

printf("%05d,",ad_data);
printf("%04d,",it_cnt);
printf("Yr");
up_date = 0;
IRQC = 0; //
}
sleep(50);
}
XPW &= ~0x8; //Disable EXT TRG
IPRA &= ~0x00f0;
}
int main(int argc, char **argv) {
int year, month, week, day, hour, min, sec;
int i, value;
int n;
unsigned char c;

PAIORH |= 0x30; //PA21,PA20:out

//XPA = 0x55;
//XPB = 0xaa;

BCR1 &= ~0x8; //A3SZ = 0 CS3:8bitbus for PC/104
PACRL2 = ((PACRL2 & 0x3fff) | 0x8000) ; //PA7MD = 2 CS3 出力 PIN

init_irq20(); //interrupt for sh2

//XPW = 0x3; //CPLD IO OUT

adin_irq0(); //interrupt external convert
adin_dif0(); //differential
adin_se0(); //Single End
}

```



## 改訂履歷

2011/10/15 初版