

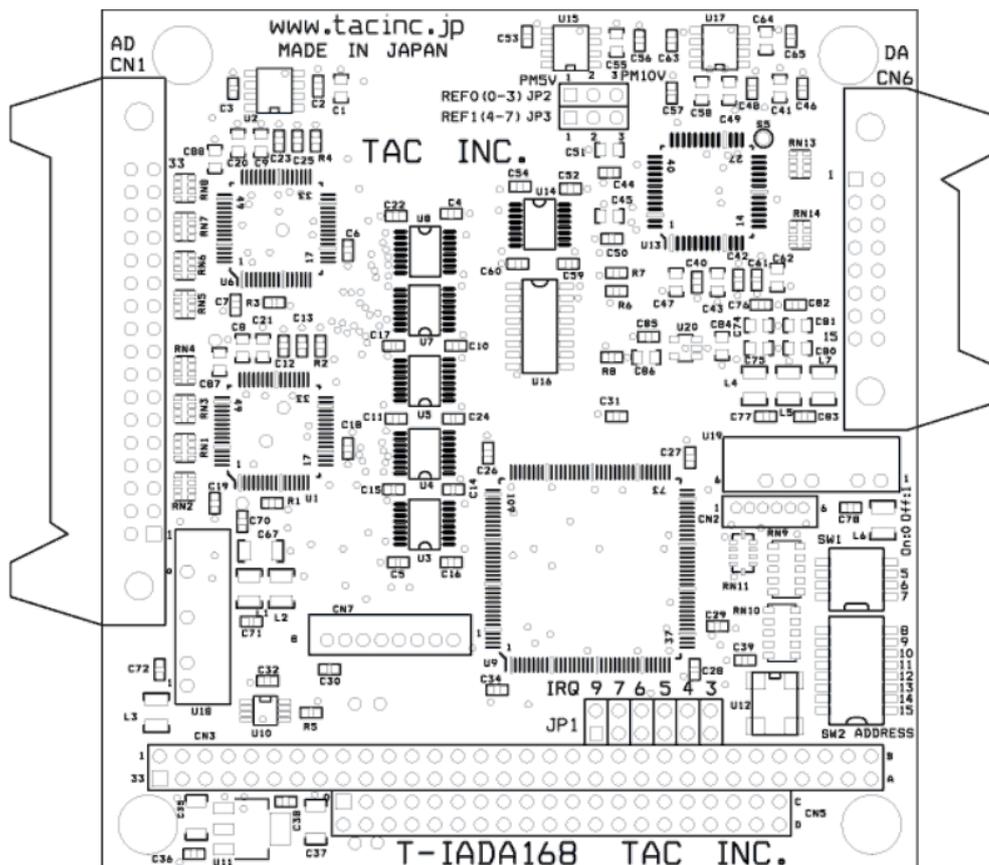
T-IADA168 rev. 1.0

PC/104バスシリーズ

絶縁16ビットAD 16ch

絶縁16ビットDA 8ch

取扱説明書



●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-IADA168 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-IADA168 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

- AD 変換 LSI アナログデバイセズ AD7606 × 2 個
16ビット分解能 16ch 同時サンプリング
デジタル・フィルタによるオーバ・サンプリング機能
変換時間 4.15 μ sec オーバーサンプリングオフ時
315 μ sec オーバーサンプリングx64(最大)
シリアル→パラレル変換時間 4.75 μ sec / ch
2.5V 基準電圧 ADR4525ARZ
入力範囲 $\pm 5V$, $\pm 10V$ ソフトで設定
入力インピーダンス: 1M Ω
変換開始ソフトウェアのみ
変換終了割り込み可能
デジタル回路と絶縁、DA と絶縁。CH 間は非絶縁。

- DA 変換 LSI アナログデバイセズ AD5362 使用 合計 8CH (オプション)
16ビット分解能
ソフトでゲインとオフセットを、独立に調節可能。
基準電圧 アナログデバイセズ ADR4525ARZ, ADR4550ARZ 使用
CPLD でパラレル→シリアル変換 27MHz パラレル→シリアル変換時間は 0.89 μ sec
出力範囲 4CH 毎に $\pm 5V$, $\pm 10V$ をジャンパーで切り換え
電源 ON で出力 0V
デジタル回路と絶縁、AD と絶縁。CH 間は非絶縁。

- PC104 バス 8ビット専用

2. IO アドレスの設定

ベース I/O アドレス(以下 BIO)は SW1 と SW2 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	
SW2番号	1	2	3	4	5	6	7	8				
SW1番号									1	2	3	
スイッチ状態	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	ON	ON	ON

出荷時の状態

3. レジスタ一覧

アドレス	読み込み	書き込み
BIO + 00H	AD 変換値 V4 下位 8 ビット	
BIO + 01H	AD 変換値 V4 上位 8 ビット	
BIO + 02H	AD 変換値 V3 下位 8 ビット	
BIO + 03H	AD 変換値 V3 上位 8 ビット	
BIO + 04H	AD 変換値 V2 下位 8 ビット	
BIO + 05H	AD 変換値 V2 上位 8 ビット	
BIO + 06H	AD 変換値 V1 下位 8 ビット	
BIO + 07H	AD 変換値 V1 上位 8 ビット	
BIO + 08H	AD 変換値 V8 下位 8 ビット	
BIO + 09H	AD 変換値 V8 上位 8 ビット	
BIO + 0AH	AD 変換値 V7 下位 8 ビット	
BIO + 0BH	AD 変換値 V7 上位 8 ビット	
BIO + 0CH	AD 変換値 V6 下位 8 ビット	
BIO + 0DH	AD 変換値 V6 上位 8 ビット	
BIO + 0EH	AD 変換値 V5 下位 8 ビット	
BIO + 0FH	AD 変換値 V5 上位 8 ビット	
BIO + 10H	Bit2:SPI Busy, Bit1:AD U6 Busy, Bit0:AD U1 Busy	AD 変換開始 Bit1:AD U1 , Bit0:AD U6
BIO + 11H	AD 入力レンジ、フィルタ	AD 入力レンジ、フィルタ
BIO + 12H		AD シリアル→パラレル読み込み選択
BIO + 13H		AD シリアル→パラレル変換開始
BIO + 18H		DA データ D7 - D0
BIO + 19H		DA データ D15 - D8
BIO + 1AH		DA データ D23 - D16
BIO + 1BH		DA パラレル→シリアル変換開始
BIO + 1CH		DA LDAC
BIO + 1DH	DA Bit1:SPI Busy Bit0:DA Busy	
BIO + 1EH		DA CLR

5. AD コンバータ

5. 1 AD コンバータ概要

AD コンバータはアナログデバイセズAD7606を2個使用しています。絶縁の為にシリアルモードで使用しております。一度に16CH同時サンプリングが可能です。読み込みはシリアル→パラレル変換が必要です。シリアル→パラレル変換は8CH分まとめて変換します。変換時間は4.75μsec。

基本的な流れは

入力レンジ設定、フィルタ設定後

AD変換開始→AD変換待→読み込みAD選択→シリアル・パラレル変換開始→変換待ち→8ビット単位で変換値を読み込み。

変換値は2の補数です。

例 -10V 8000h
0V 0000h
+10V 7FFFh

5. 2 入力レンジ フィルタ設定 BIO + 11H

ADの入力レンジ±5Vまたは±10Vの設定、オーバーサンプリングフィルタの設定はBIO + 11Hで設定します。書き込みビットの割り当ては

ビット位置		AD 番号
7(MSB)	レンジ 1:±10V 0:±5V	U1
6	OS2(MSB) オーバーサンプリング	U1
5	OS1	U1
4	OS0(LSB)	U1
3	レンジ 1:±10V 0:±5V	U6
2	OS2(MSB) オーバーサンプリング	U6
1	OS1	U6
0(LSB)	OS0(LSB)	U6

OS[2:0]は0-6の設定が可能です。7は不可。

Table 9. Oversample Bit Decoding

OS[2:0]	OS Ratio	SNR 5 V Range (dB)	SNR 10 V Range (dB)	3 dB BW 5 V Range (kHz)	3 dB BW 10 V Range (kHz)	Maximum Throughput CONVST Frequency (kHz)
000	No OS	89	90	15	22	200
001	2	91.2	92	15	22	100
010	4	92.6	93.6	13.7	18.5	50
011	8	94.2	95	10.3	11.9	25
100	16	95.5	96	6	6	12.5
101	32	96.4	96.7	3	3	6.25
110	64	96.9	97	1.5	1.5	3.125
111	Invalid					

Rev. C | Page 29 of 36

例 U1 ±5V、オーバーサンプリングレシオ64

U6 ±10V、オーバーサンプリング無しの設定は = 68H

設定を変更後は2, 3回ダミーのAD変換が必要です。

5. 3 AD変換 BIO + 10H

BIO + 10Hにデータを書き込むとAD変換を開始します。

Bit1:U1 Bit0:U6 に割り当てています。

例

BIO + 10Hに03Hを書き込むとU1, U6同時にAD変換開始。

BIO + 10Hに01Hを書き込むとU6のみAD変換開始。

BI0 + 10H を読むことで変換中の状態を知ることができます。
 変換と同様 Bit1:U1 Bit0:U6 に割り当てています。
 1の時、変換中です。

5. 4 AD 変換値 読み込み BI0 + 00H - 0FH

AD 変換の終了後、シリアル→パラレル変換、データの読み出しが可能です。
 シリアル→パラレル変換は一度に8CH分のみ変換です。16CH分読み出す場合は
 2回シリアル→パラレル変換が必要です。

U6 または U1 どちらのデータを読み出すかを
 BI0 + 12H に設定します。0 を書き込むと U6, 1 を書き込むと U1 の選択になります。

BI0 + 10H の Bit2 が 1 の時はシリアル→パラレル変換中です。

シリアル→パラレル変換後、BI0 + 00H - 0FH から変換データを読み込みます。

5. 5 AD 入力コネクタ

AD 入力コネクタ CN1 ピン配列 HIROSE HIF3BA-34PA-2.54DS(71) または互換品

ピン番号	信号名	ピン番号	信号名	AD コンバータ 番号
1	未使用	2	未使用	
3	V8 GND	4	V8	U1
5	V7 GND	6	V7	U1
7	V6 GND	8	V6	U1
9	V5 GND	10	V5	U1
11	V4 GND	12	V4	U1
13	V3 GND	14	V3	U1
15	V2 GND	16	V2	U1
17	V1 GND	18	V1	U1
19	V8 GND	20	V8	U6
21	V7 GND	22	V7	U6
23	V6 GND	24	V6	U6
25	V5 GND	26	V5	U6
27	V4 GND	28	V4	U6
29	V3 GND	30	V3	U6
31	V2 GND	32	V2	U6
33	V1 GND	34	V1	U6

7. サンプルプログラム

OS : MES 2.6
CPU : T-SH2MB
割り込み未使用。

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

//T-IADA168 Base address = 0x300
#define ADL4 (*(volatile unsigned char *)0xc00300)
#define ADU4 (*(volatile unsigned char *)0xc00301)
#define ADL3 (*(volatile unsigned char *)0xc00302)
#define ADU3 (*(volatile unsigned char *)0xc00303)
#define ADL2 (*(volatile unsigned char *)0xc00304)
#define ADU2 (*(volatile unsigned char *)0xc00305)
#define ADL1 (*(volatile unsigned char *)0xc00306)
#define ADU1 (*(volatile unsigned char *)0xc00307)
#define ADL8 (*(volatile unsigned char *)0xc00308)
#define ADU8 (*(volatile unsigned char *)0xc00309)
#define ADL7 (*(volatile unsigned char *)0xc0030a)
#define ADU7 (*(volatile unsigned char *)0xc0030b)
#define ADL6 (*(volatile unsigned char *)0xc0030c)
#define ADU6 (*(volatile unsigned char *)0xc0030d)
#define ADL5 (*(volatile unsigned char *)0xc0030e)
#define ADU5 (*(volatile unsigned char *)0xc0030f)

#define AD_STS (*(volatile unsigned char *)0xc00310) //b2:spi busy b1:ad2 busy b0:ad1 busy write convert
#define AD_RANGE (*(volatile unsigned char *)0xc00311)
#define AD_SEL (*(volatile unsigned char *)0xc00312)
#define AD_SS (*(volatile unsigned char *)0xc00313) //write to start spi

#define DA_L (*(volatile unsigned char *)0xc00318)
#define DA_M (*(volatile unsigned char *)0xc00319)
#define DA_H (*(volatile unsigned char *)0xc0031a)

#define DA_SS (*(volatile unsigned char *)0xc0031b) //write to start spi
#define DA_LDAC (*(volatile unsigned char *)0xc0031c) //Latch
#define DA_STS (*(volatile unsigned char *)0xc0031d) //b1:spi busy b0:da busy
#define DA_CLR (*(volatile unsigned char *)0xc0031e) //Clr

void adin(void)
{
    char c;
    // AD_RANGE = 0x80; //ad2:10V ad1:5V
    //AD_RANGE = 0x08; //ad2:5V ad1:10V
    //AD_RANGE = 0x19; //ad2:5V os1 ad1:10V os1
    //AD_RANGE = 0x6e; //ad2:5V os6 ad1:10V os6 max 6 not 7
    //AD_RANGE = 0xe6; //ad2:10V os6 ad1:5V os6
    //AD_RANGE = 0x88; //ad2:10V ad2:10V
    AD_RANGE = 0xee; //ad2:10V ad2:10V os6 max
    //AD_RANGE = 0x88; //ad2:10V ad2:10V no os
    // AD_RANGE = 0x99; //ad2:10V ad2:10V os1 min
```

```

//Need dummy convert after change OS
AD_STS = 0x3; //AD convert 0-15ch
while( AD_STS & 0x3 );

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

//      AD_STS = 0x3; //AD convert 0-15ch
//      while( AD_STS & 0x3 );

      AD_STS = 0x1; //AD convert 0-7ch
      while( AD_STS & 0x1 );

      AD_SEL = 0x0; //select ad1
      AD_SS = 0x0; //start serial to parallel
      while( AD_STS & 0x4 ); //

      printf("%02x",ADU1);
      printf("%02x",ADL1);
      printf("%02x",ADU2);
      printf("%02x",ADL2);
      printf("%02x",ADU3);
      printf("%02x",ADL3);
      printf("%02x",ADU4);
      printf("%02x",ADL4);
      printf("%02x",ADU5);
      printf("%02x",ADL5);
      printf("%02x",ADU6);
      printf("%02x",ADL6);
      printf("%02x",ADU7);
      printf("%02x",ADL7);
      printf("%02x",ADU8);
      printf("%02x",ADL8);

      /*
      AD_SEL = 0x1; //select ad2
      AD_SS = 0x0; //start serial to parallel
      while( AD_STS & 0x4 );

      printf("%02x",ADU1);
      printf("%02x",ADL1);
      printf("%02x",ADU2);
      printf("%02x",ADL2);
      printf("%02x",ADU3);
      printf("%02x",ADL3);
      printf("%02x",ADU4);
      printf("%02x",ADL4);
      printf("%02x",ADU5);
      printf("%02x",ADL5);
      printf("%02x",ADU6);
      printf("%02x",ADL6);
      printf("%02x",ADU7);
      printf("%02x",ADL7);

```

```

printf("%02x",ADU8);
printf("%02x",ADL8);
*/

printf("¥r");
//sleep(10);
}
}
int main(int argc, char **argv) {
int i, value;
int n;
unsigned char c;

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

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

adin();

}

```

6. DA コンバータ

6. 1 DA コンバータ概要

DA コンバータはアナログデバイセズ AD5362 を 1 個使用しています。
4ch 毎に±10V レンジまたは±5V レンジの設定が可能です。
変換値はバイナリです。

例 -10V 0000h
0V 8000h
+10V FFFFh

DA のレジスタはデータレジスタ以外にオフセット、ゲインレジスタがあります。
調整が不要な場合はデータレジスタのみの使用になります。

6. 2 出力レンジ切り換え

ジャンパー No.	ジャンパー位置	レンジ
JP2	1-2	DA 0-3ch ±5V
JP2	2-3	DA 0-3ch ±10V
JP3	1-2	DA 4-7ch ±5V
JP3	2-3	DA 4-7ch ±10V

6. 3 DA 出力コネクタ CN6 ピン配列 HIROSE HIF3BA-16PA-2.54DS(71) または互換品

ピン番号	信号名	ピン番号	信号名
1	CH3	2	GND
3	CH2	4	GND
5	CH1	6	GND
7	CH0	8	GND
9	CH7	10	GND
11	CH6	12	GND
13	CH5	14	GND
15	CH4	16	GND

6. 4 DA サンプル

```

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

//T-IADA168 Base address = 0x300

#define DA_L    (*(volatile unsigned char *)0xc00318)
#define DA_M    (*(volatile unsigned char *)0xc00319)
#define DA_H    (*(volatile unsigned char *)0xc0031a)

#define DA_SS    (*(volatile unsigned char *)0xc0031b) //write to start spi
#define DA_LDAC  (*(volatile unsigned char *)0xc0031c) //Latch
#define DA_STS    (*(volatile unsigned char *)0xc0031d) //b1:spi busy b0:da busy
#define DA_CLR    (*(volatile unsigned char *)0xc0031e) //Clr

void set_da(int x, int ch, unsigned int data) {
    unsigned char xd;

    if(ch < 4)        xd =  (8 + ch );
    else              xd =  (16 + ch - 4 );

    if(x == 0)        xd |= 0xc0; //data register
    else if(x == 1)   xd |= 0x80; //offset register
    else if(x == 2)   xd |= 0x40; //gain register

    DA_H = xd;
    DA_M = ((data >> 8) & 0xff);
    DA_L = data & 0xff ; //

    DA_SS = 0x55; //start spi
    while( (DA_STS & 0x3) != 3 ) ; //waiting spi busy DA
}

void test_da(void) {
    unsigned int data = 0;
    int i;
    unsigned char c;

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

        set_da(0, 0, 0x2000); //ch0 data
        set_da(0, 1, 0x4000 );
        set_da(0, 2, 0x6000 );
        set_da(0, 3, 0x8000 );
        set_da(0, 4, 0xa000 );
        set_da(0, 5, 0xc000 );
    }
}

```

```

        set_da(0, 6, 0xe000 );
        set_da(0, 7, 0xffff );

        // set_da(1, 3, data ); //offset
        set_da(2, 3, data ); //gain

        printf("%04x %n,", data);
        ++data;
    }
}
int main(int argc, char **argv) {
    int i, value;
    int n;
    unsigned char c;

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

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

    test_da();
}

```

改訂履歷

2014/10/23 初版
2016/02/09 DA 追加