

第七章 調整

7-1 調整

各チャンネルのアナログ調整は以下1～4項について順番に行います。

1：基準電圧調整

次の表に従って基準電圧を調整して下さい

トリマ番号	測定ポイント	調整値 (V)
VR25	T7-T8	+10.020

2：オフセット

表1から対応するレンジのオフセット値を選び、この値になるように各チャンネルに対応するトリマーを調整する。この時、各チャンネルとも2ヶ所の調整トリマーがありますが、次の表に示す通り上段、下段の順番にて調整して下さい。
又、各チャンネルに対する設定コード値は000 (HEX) です。

チャンネル番号	トリマ番号	測定ポイント	調整値 (V)
0	VR 2	T17-T 9	0.0000
0	VR 1	CN1:21- 1	表1参照
1	VR 5	T17-T10	0.0000
1	VR 4	CN1:23- 3	表1参照
2	VR 8	T17-T11	0.0000
2	VR 7	CN1:25- 5	表1参照
3	VR11	T17-T12	0.0000
3	VR10	CN1:27- 7	表1参照
4	VR14	T17-T13	0.0000
4	VR13	CN1:29- 9	表1参照
5	VR17	T17-T14	0.0000
5	VR16	CN1:31-11	表1参照
6	VR20	T17-T15	0.0000
6	VR19	CN1:33-13	表1参照
7	VR23	T17-T16	0.0000
7	VR22	CN1:35-15	表1参照

3 : ゲイン調整

表1から対応するレンジのゲイン値を選び、この値になるように各チャンネルに対応するトリマーを調整する。
又、各チャンネルに対する設定コード値はFFF (HEX) です。

チャンネル番号	トリマ番号	測定ポイント	調整値 (V)
0	VR 3	CN1 : 21 - 1	表1参照
1	VR 6	CN1 : 23 - 3	表1参照
2	VR 9	CN1 : 25 - 5	表1参照
3	VR 12	CN1 : 27 - 7	表1参照
4	VR 15	CN1 : 29 - 9	表1参照
5	VR 18	CN1 : 31 - 11	表1参照
6	VR 21	CN1 : 33 - 13	表1参照
7	VR 24	CN1 : 35 - 15	表1参照

4 : センター・ポイント確認

各チャンネルに対して設定コード800 (HEX) をセットした時、表1から対応するセンター値を選び、アナログ出力がこの値になっていることを確認します。
もし、規定値外であれば対応するチャンネルのオフセット、ゲインについて再度調整を行います。

チャンネル番号	測定ポイント	確認値 (V)
0	CN1 : 21 - 1	表1参照
1	CN1 : 23 - 3	表1参照
2	CN1 : 25 - 5	表1参照
3	CN1 : 27 - 7	表1参照
4	CN1 : 29 - 9	表1参照
5	CN1 : 31 - 11	表1参照
6	CN1 : 33 - 13	表1参照
7	CN1 : 35 - 15	表1参照

表1 :

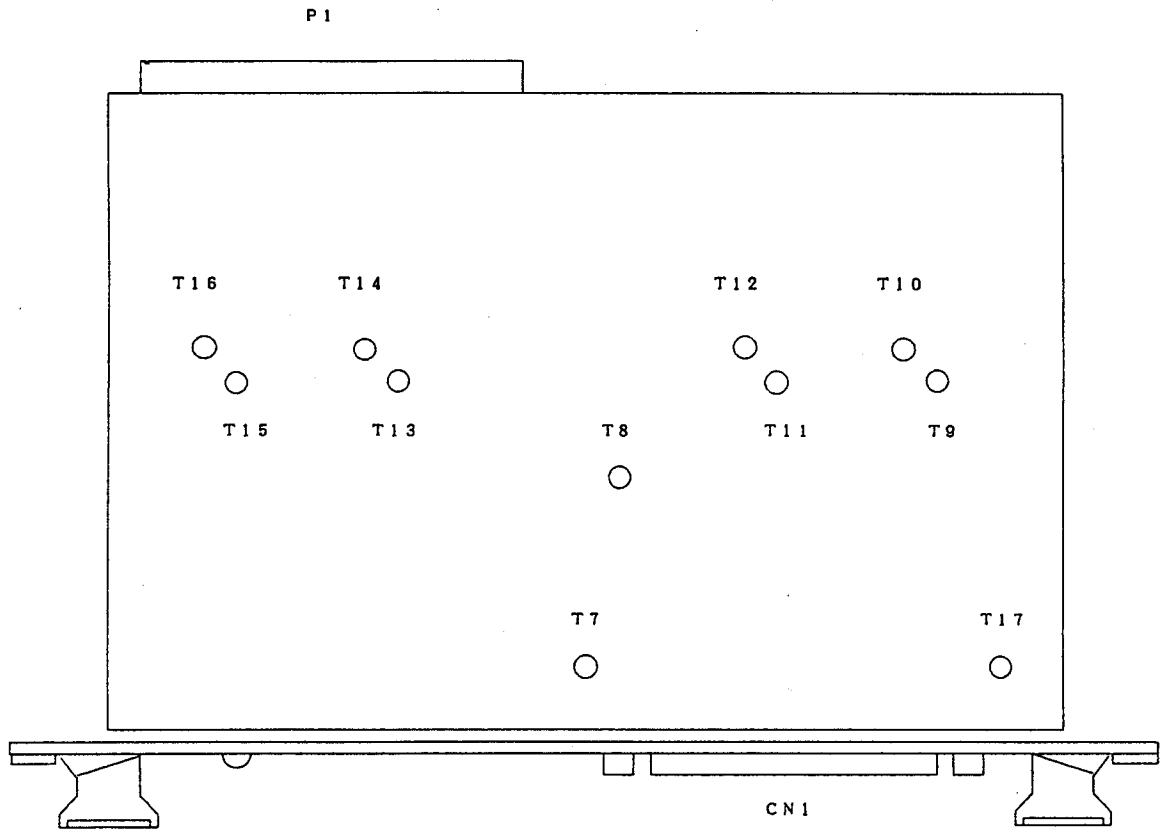
出力レンジ	オフセット (V)	ゲイン (V)	センター
± 10.0V	-10.0000	+9.9951	0±2.44 (mV)
± 5.0V	- 5.0000	+4.9975	0±1.22 (mV)
± 2.5V	- 2.5000	+2.4987	0±610 (μV)
0-10.0V	0.0000	+9.9975	0±1.22 (mV)
0- 5.0V	0.0000	+4.9987	0±610 (μV)
0- 2.5V	0.0000	+2.4993	0±305 (μV)

注意) 表記について

測定ポイント: Txx-Txxは左側GND/右側+側
(ボード上のテスト・ピン番号)

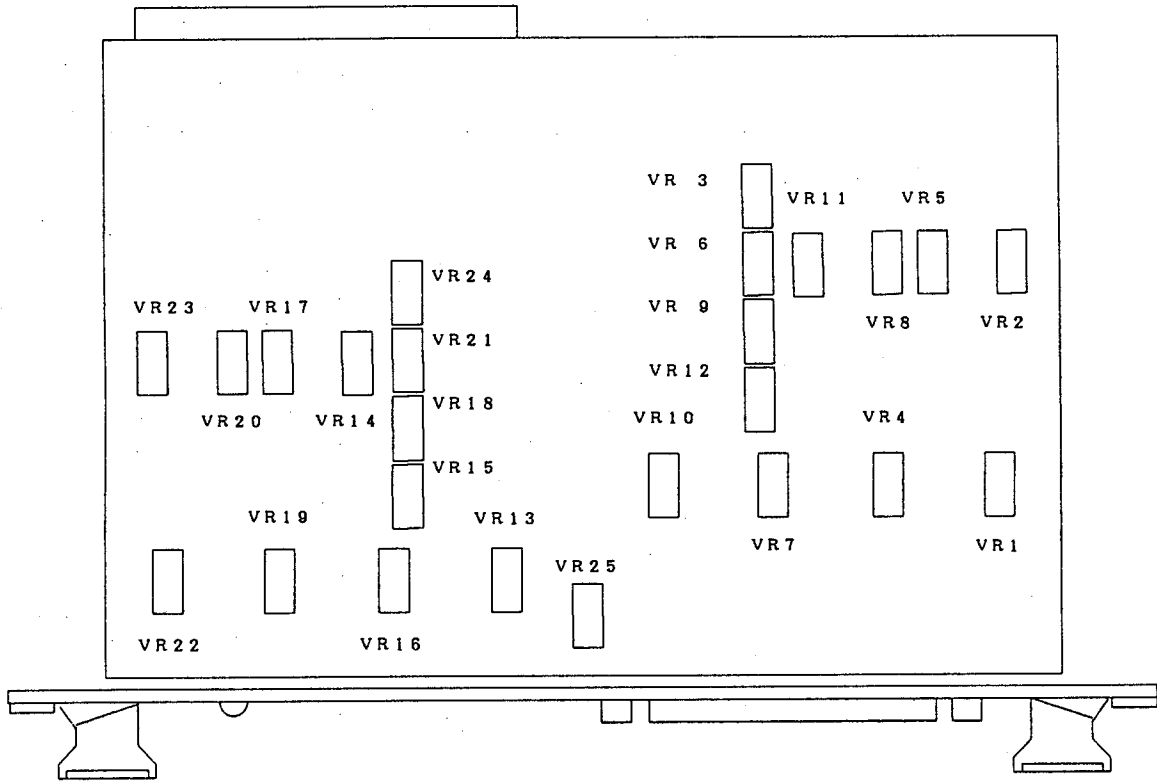
: CN1: xx-xxは左側GND/右側+側
(フロント側コネクタCN1のピン番号)

7-2 テスト・ピン配置図



部品面側

7-3 トリマー配置図



部品面側

第八章 サンプル・プログラム

8-1 サンプル・プログラム

8-1-1 チャンネル0~7にてフル・スケールでの三角波出力

データ・レジスタ0~7操作時：

```

/*****
/*      PVME-323 SAMPLE PROGRAM      */
/*          USE REGISTER DATA REGISTER 0-7      */
/*          INTERNIX. INC      */
*****/
#include      <stdio.h>
#define      BASE 0xfc4e0000
/*-----*/
/* main      */
/*-----*/
main()
{
    unsigned char ch;
    unsigned short data, time;
    unsigned long time;

    time = 0xffff;      /* wait time */

    on();      /* ch0-7 output enable */

    while(1)
    {
        for(i=0; i<4096; i++)
        {
            for(ch=0; ch<8; ch++)
            {
                out(ch, data);      /* ch0-7 out */
                wait(time);      /* wait */
            }
        }

        for(i=4096; i>0; i--)
        {
            for(ch=0; ch<8; ch++)
            {
                out(ch, data);      /* ch0-7 out */
                wait(time);      /* wait */
            }
        }
    }
}

```



```

on()
{
    unsigned char *creg0;
    creg0 = (unsigned char *) (BASE+0x19);
    *creg0 = 0x01;          /* output enable */
}

out(ch, data)
    unsigned char ch;
    unsigned short data;
    {
        unsigned short *dr0, *dr1, *dr2, *dr3, *dr4, *dr5, *dr6, *dr7, *syr;
        dr0 = (unsigned short *) (BASE+0x30); /* data register 0 */
        dr1 = (unsigned short *) (BASE+0x32); /* data register 1 */
        dr2 = (unsigned short *) (BASE+0x34); /* data register 2 */
        dr3 = (unsigned short *) (BASE+0x36); /* data register 3 */
        dr4 = (unsigned short *) (BASE+0x38); /* data register 4 */
        dr5 = (unsigned short *) (BASE+0x3a); /* data register 5 */
        dr6 = (unsigned short *) (BASE+0x3c); /* data register 6 */
        dr7 = (unsigned short *) (BASE+0x3e); /* data register 7 */
        syr = (unsigned short *) (BASE+0x40); /* synchronize register */

        data = (data & 0x0fff);          /* 12 bit mask */
        switch(ch)
        {
            case 0:
                *dr0 = data; break;      /* channel 0 */
            case 1:
                *dr1 = data; break;      /* channel 1 */
            case 2:
                *dr2 = data; break;      /* channel 2 */
            case 3:
                *dr3 = data; break;      /* channel 3 */
            case 4:
                *dr4 = data; break;      /* channel 4 */
            case 5:
                *dr5 = data; break;      /* channel 5 */
            case 6:
                *dr6 = data; break;      /* channel 6 */
            case 7:
                *dr7 = data; break;      /* channel 7 */
            case 8:
                *syr = data; break;      /* channel 0-7 */
            default:
                printf("¥n¥ ch number error");
        }
    }
}

```

```

wait(time)
unsigned long time;
{
for (y=0;i<time;i++)
{
/* time count */
}
}

off()
{
unsigned char *creg0;
creg0 = (unsigned char *) (BASE+0x19);
*creg0 = 0x00; /* out disable */
}

/*-----*/
/* e.o.f */
/*-----*/

```

注：1) ベース・アドレスはBASEにてfc4e0000に設定されていますので
アプリケーションに合わせ御変更願います。

2) wait () による時間はソフト・ウェア・タイマの為、ご使用になるCPU
により異なりますので、ユーザーにて御変更願います。

8-1-2 チャンネル0~7にてフル・スケールでの三角波出力

シンクロナイズ・レジスタ操作時:

```

/*****/
/*      PVME-323 SAMPLE PROGRAM      */
/*      USE REGISTER SYNCHRONIZE      */
/*      INTERNIX. INC                  */
/*****/
#include      <stdio.h>
#define      BASE      0xfc4e0000
/*-----*/
/* main */
/*-----*/
main()
{
  unsigned char ch;
  unsigned short data, time;
  unsigned long time;

  ch = 8;          /* synchronize channel */

  time = 0xffff;  /* wait time */

  on();           /* ch0-7 output enable */

  while(1)
  {
    for (data=0; data<4096; data++)
    {
      out(ch, data); wait (time);      /* ch0-7 out & write wait */
    }
    for (data=4096; data>0; data--)
    {
      out(ch, data); wait (time);      /* ch0 out & write wait */
    }
  }
}

```

```

on()
{
    unsigned char *creg0;
    creg0 = (unsigned char *) (BASE+0x19);
    *creg0 = 0x01;                /* output enable */
}

out(ch, data)
    unsigned char ch;
    unsigned short data;
    {
        unsigned short *dr0, *dr1, *dr2, *dr3, *dr4, *dr5, *dr6, *dr7, *syr;
        dr0 = (unsigned short *) (BASE+0x30); /* data register 0 */
        dr1 = (unsigned short *) (BASE+0x32); /* data register 1 */
        dr2 = (unsigned short *) (BASE+0x34); /* data register 2 */
        dr3 = (unsigned short *) (BASE+0x36); /* data register 3 */
        dr4 = (unsigned short *) (BASE+0x38); /* data register 4 */
        dr5 = (unsigned short *) (BASE+0x3a); /* data register 5 */
        dr6 = (unsigned short *) (BASE+0x3c); /* data register 6 */
        dr7 = (unsigned short *) (BASE+0x3e); /* data register 7 */
        syr = (unsigned short *) (BASE+0x40); /* synchronize register */

        data = (data & 0x0fff);                /* 12 bit mask */
        switch(ch)
        {
            case 0:
                *dr0 = data; break;            /* channel 0 */
            case 1:
                *dr1 = data; break;            /* channel 1 */
            case 2:
                *dr2 = data; break;            /* channel 2 */
            case 3:
                *dr3 = data; break;            /* channel 3 */
            case 4:
                *dr4 = data; break;            /* channel 4 */
            case 5:
                *dr5 = data; break;            /* channel 5 */
            case 6:
                *dr6 = data; break;            /* channel 6 */
            case 7:
                *dr7 = data; break;            /* channel 7 */
            case 8:
                *syr = data; break;            /* channel 0-7 */
            default:
                printf("%#n* ch number error");
        }
    }
}

```

```

wait (time)
  unsigned long time;
  {
  for (y=0;i<time;i++)
  {
  }
  }
  }
  }

off ()
{
  unsigned char *creg0;
  creg0 = (unsigned char *) (BASE+0x19);
  *creg0 = 0x00;
  }

/*-----*/
/*  e.o.f  */
/*-----*/

```

注：1) ベース・アドレスはBASEにてfc4e0000に設定されていますのでアプリケーションに合わせ御変更願います。

2) wait () による時間はソフト・ウェア・タイマの為、ご使用になるCPUにより異なりますので、ユーザーにて御変更願います。

8-1-3 関数概要

1) out (ch, data) : 各チャンネルのアナログ出力

引き数: ch	チャンネル番号	0:チャンネル	0
			1:チャンネル	1
			2:チャンネル	2
			3:チャンネル	3
			4:チャンネル	4
			5:チャンネル	5
			6:チャンネル	6
			7:チャンネル	7
			8:シンクロナイズ (チャンネル0-7)	同時出力

data.....コード設定値 12ビットD/A変換コード

返 値:なし

エラー時: 指定外チャンネル番号入力時、” ch number error ” 表示

2) on () : 出力イネーブル

引き数:なし

返 値:なし

3) off () : 出力ディセーブル

引き数:なし

返 値:なし

4) wait (time) : ソフト・ウェア・タイマー

引き数: 符号なしロング型整数

返 値:なし