

第 9 章 アナログ調整

9-1 調整方法

PVME-303はアナログ入力について6レンジが用意されておりますが、出荷時には±10[V]レンジにて正確に調整されています。

したがって、他のレンジにて御使用の際は所定のジャンパーにて希望のレンジを設定後、再度調整が必要となります。

調整については次の手順に従い行います。

1) ボードの各設定事項

項目	設定値	部品番号
アドレス モード	OR (特権及び非特権)	LD 7
ベース アドレス	FC4B0000h	LD 1-LD 6
インターバル タイマ	カスケード接続	JP 17
チャンネル・ナンバー	有	DP 1-8
FIFO アクセス	ワード	DP 1-9
A/D モード	モード1 (マスタ)	DP 1-7

* 出荷時設定

2) 入力レンジ

チャンネル0-7のユニポーラ/バイポーラ・レンジ設定はジャンパー (JP1-16)にて行います。

各チャンネルとジャンパーの対応及びフォーマットは次の通りです。

チャンネル番号	ユニポーラ/バイポーラ	レンジ
0	JP 9	JP 1
1	JP 10	JP 2
2	JP 11	JP 3
3	JP 12	JP 4
4	JP 13	JP 5
5	JP 14	JP 6
6	JP 15	JP 7
7	JP 16	JP 8

ユニポーラ／バイポーラ設定フォーマット

JP9-JP16共通

3 1

○	○
○	○

3	1	ユニポーラ／バイポーラ
●	○	ユニポーラ
○	●	バイポーラ

レンジ設定フォーマット

7 5 3 1 (ユニポーラ時)

○	○	○	○
○	○	○	○

7	5	3	1	レンジ [V]
○	○	○	●	0 - 10.0
○	●	○	○	0 - 5.0
●	○	○	○	0 - 2.5

(バイポーラ時)

7	5	3	1	レンジ [V]
○	○	●	○	± 10.0
○	○	○	●	± 5.0
○	●	○	○	± 2.5

3) 調整プログラム (TEST303:C)

付随した調整プログラム (C言語記述) をシステムに入力後、コンパイルにて実行して下さい。

又、このときベース・アドレスを変換する場合はファイル (ADRS303:H) のベース・アドレス部を希望の値に変更願います。

4) 調整プログラム実行

調整プログラム実行後、モニタ上にチャンネル・ナンバ/変換データが次の様に表示されます。

各チャンネルの調整はこのモニタに表示されたデータを確認しながら所定のトリマを調整することで行います。

注) モニタはVT52モードに設定とします。

```

*****
* OFFSET & GAIN ADJUST *
*****

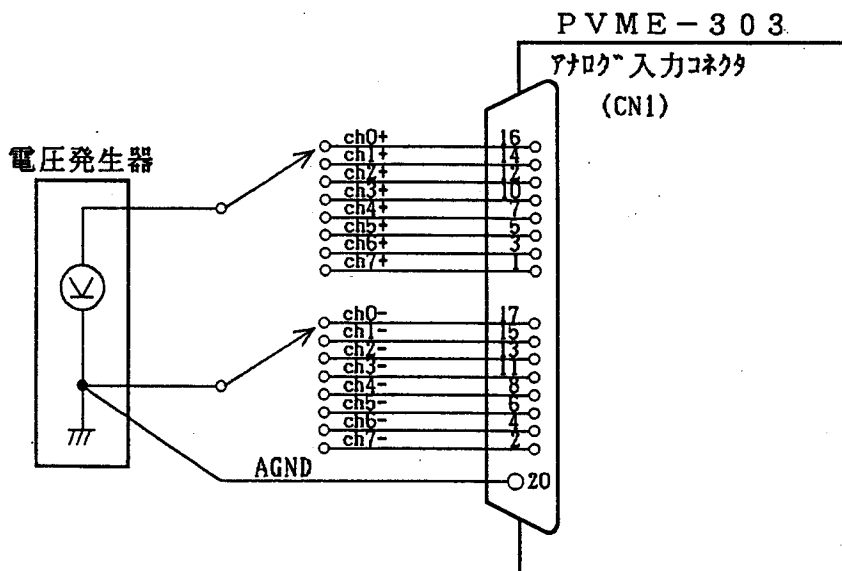
(CHANNEL) (A/D DATA)
ch:0      data:800
ch:1      data:801
ch:2      data:801
ch:3      data:800
ch:4      data:800
ch:5      data:800
ch:6      data:801
ch:7      data:801 変換データ (HEX表示)

*now CHO-7 ANALOG ADJUSTING*
    
```

モニタ表示例

5) 調整電圧

調整電圧はアナログ入力コネクタ (CN1) の各チャンネルに対応したピンに接続して下さい。



接続図

6) 調整

① オフセット調整

各レンジごとに対応した1/2LSB電圧を調整するチャンネルに入力後、モニタ上に表示されるA/D変換値が000~001にて半々に表示するようにチャンネルに対応したトリマを調整します。

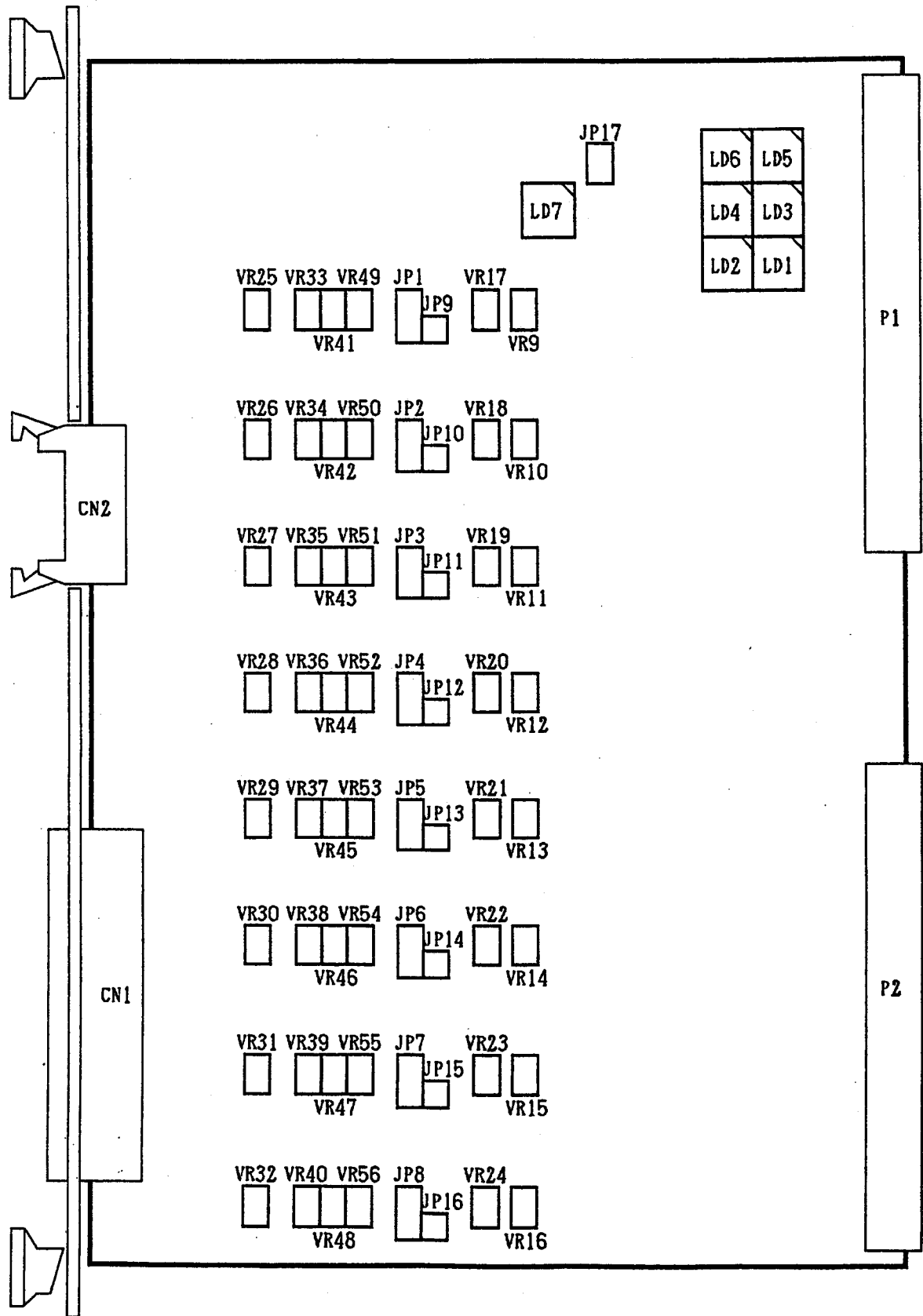
レンジ	1/2LSB電圧[mV]	トリマ (部品番号)							
		ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7
0-2.5[V]	0.30517	VR17	VR18	VR19	VR20	VR21	VR22	VR23	VR24
0-5.0[V]	0.61035								
0-10.0[V]	1.22070								
±2.5[V]	0.61035	VR9	VR10	VR11	VR12	VR13	VR14	VR15	VR16
±5.0[V]	1.22070								
±10.0[V]	2.44140								

② ゲイン調整

各レンジごとに対応するFSR (フルスケール) -1.5LSB電圧を調整するチャンネルに入力後、モニタ上に表示されるA/D変換値がFFE~FFFにて半々に表示するようにチャンネルに対応したトリマを調整します。

レンジ	FSR-1.5LSB 電圧[mV]	トリマ (部品番号)							
		ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7
0-2.5[V]	2.49908	VR25	VR26	VR27	VR28	VR29	VR30	VR31	VR32
0-5.0[V]	4.99816	VR33	VR34	VR35	VR36	VR37	VR38	VR39	VR40
0-10.0[V]	9.99633	VR41	VR42	VR43	VR44	VR45	VR46	VR47	VR48
±2.5[V]	2.49816	VR33	VR34	VR35	VR36	VR37	VR38	VR39	VR40
±5.0[V]	4.99633	VR41	VR42	VR43	VR44	VR45	VR46	VR47	VR48
±10.0[V]	9.99267	VR49	VR50	VR51	VR52	VR53	VR54	VR55	VR56

9-2 部品位置



9-3 調整プログラム (TEST303:C / ADRS303:H)

```

/*****
/*
/*          PVME-303 OFFSET/GAIN ADJUSTING PROGRAM  */
/*          PROGRAM NAME : TEST303:C                */
/*          Written By Internix.inc                */
/*****
/*-----*/
/* INCLUDE          */
/*-----*/
#include          <MAST303:H>      /* base fc4b0000 */
#include          <stdio.h>
/*-----*/
/* main            */
/*-----*/
main()
{
  init();          /* master initializ */
  clrclr();       /* CRT clear */
  tx0();          /* 'offset/gain adjust' */
  tx1();          /* 'a/d data' */
  while(1)
  {
    adjust();     /* offset/gain adjust */
  }
}
/*-----*/
/* initializ master */
/*-----*/
init()
{
  unsigned char *tr0,*tr1,*cnr,*cr4,*cr5,*cr6;
  tr0 = (unsigned char *) (TR0);
  tr1 = (unsigned char *) (TR1);
  cnr = (unsigned char *) (CNR);
  cr4 = (unsigned char *) (CR4);
  cr5 = (unsigned char *) (CR5);
  cr6 = (unsigned char *) (CR6);
  *cr4 = 0x34;    /* intervaltimer1 */
  *cr4 = 0x74;    /* intervaltimer2 */
  *cr4 = 0xba;    /* loopcounter */
  *tr0 = 0x08;    /* intervaltimer:1 us */
  *tr0 = 0x00;
  *tr1 = 0x0a;    /* intervaltimer:10 us */
  *tr1 = 0x00;
  *cnr = 0x01;    /* loopcounter:1 */
  *cnr = 0x00;
  *cr5 = 0x02;    /* yuugen */
  *cr6 = 0x07;    /* channel:8 */
}
/*-----*/
/* scanning start */
/*-----*/
getad()
{
  unsigned char *cr7,*cr8;
  cr7 = (unsigned char *) (CR7);
  cr8 = (unsigned char *) (CR8);
  ffrs();        /* fifo reset */
  *cr7 = 0x01;   /* triger enable */
  *cr8 = 0xff;   /* triger on */
}
/*-----*/
/* fifo reset */
/*-----*/
ffrs()
{
  unsigned char *cr9;
  cr9 = (unsigned char *) (CR9);
  *cr9 = 0xff;   /* fifo reset */
}
/*-----*/
/* offset/gain adjust */
/*-----*/
adjust()
{
  getad();      /* scanning start */
  done();       /* scan done ? */
  addt();       /* scan data display */
}
/*-----*/
/* scan done check */

```

```

/*-----*/
done()
{
  unsigned char *cr10;
  unsigned char s;
  unsigned char cmd0,cmd1,cmd2,cmd3;
  cr10 = (unsigned char *) (CR10);

  cmd0 = 0x1b;          /* ESC */
  cmd1 = 0x59;          /* Y */
  cmd2 = (15 + 0x20);   /* 15 line */
  cmd3 = (1 + 0x20);   /* 1 calam */

  for(s = 0x08;s == 0x08;)
  {
    s = (*cr10 & 0x08);
    printf("%c%c%c%c",cmd0,cmd1,cmd2,cmd3);
    printf("\n*now CHO-7 ANALOG ADJUSTING*");
  }
}
/*-----*/
/* scan data ch/data */
/*-----*/
addt()
{
  unsigned short *dr0;
  unsigned char i;
  unsigned char cmd0,cmd1,cmd2,cmd3,cmd4;
  unsigned short data,ch,a;

  cmd0 = 0x1b;          /* ESC */
  cmd1 = 0x59;          /* Y */
  cmd2 = (7 + 0x20);    /* 7 line */
  cmd3 = (4 + 0x20);    /* 4 calum */
  cmd4 = (15 + 0x20);   /* 15 calum */

  dr0 = (unsigned short *) (DR0);
  for(i = 0;i < 8;i++)
  {
    data = *dr0;
    ch = ((data & 0xf000)>>12);
    data = (data & 0x0fff);
    printf("%c%c%c%c",cmd0,cmd1,cmd2,cmd3);
    printf("%d",ch);
    printf("%c%c%c%c",cmd0,cmd1,cmd2,cmd4);
    printf("%3x",data);
    cmd2++;
  }
}
/*-----*/
/* crt clear */
/*-----*/
crtclr()
{
  unsigned char cmd0,cmd1,cmd2,cmd3;

  cmd0 = 0x1b;          /* ESC */
  cmd1 = 0x59;          /* Y */
  cmd2 = (0 + 0x20);    /* 0 line */
  cmd3 = (0 + 0x20);    /* 0 calam */

  printf("%c%c%c%c",cmd0,cmd1,cmd2,cmd3); /* 1 line/ 1 calam */

  cmd0 = 0x1b;          /* ESC */
  cmd1 = 0x4a;          /* J */

  printf("%c%c",cmd0,cmd1); /* CRT crear */
}
/*-----*/
/* messege 0,1 */
/*-----*/
tx0()
{
  printf("\n*****");
  printf("\n* OFFSET & GAIN ADJUST *");
  printf("\n*****");
}
tx1()
{
  printf("\n");
}

```



```
printf("\n(CHANNEL) (A/D DATA)");
printf("\n");
printf("\nch:      data:");
printf("\nch:      data:");
printf("\nch:      data:");
printf("\nch:      data:");
printf("\nch:      data:");
printf("\nch:      data:");
printf("\nch:      data:");
printf("\nch:      data:");
}
/*****
/* EOF      */
*****/
```

```

/*****/
/*
/*          PVME-303 ADDRESSING          */
/*          PROGRAM NAME : ADRS303:H     */
/*          Written By Internix.inc      */
/*****/

#define      BASE      0xfc4b0000      /* base adrs */
#define      CR0       BASE+0x01      /* irq ef */
#define      CR1       BASE+0x03      /* irq hf */
#define      CR2       BASE+0x05      /* irq ff */
#define      CR3       BASE+0x07      /* irq done */
#define      VR0       BASE+0x09      /* vct ef */
#define      VR1       BASE+0x0b      /* vct hf */
#define      VR2       BASE+0x0d      /* vct ff */
#define      VR3       BASE+0x0f      /* vct done */
#define      TR0       BASE+0x11      /* interval 1 */
#define      TR1       BASE+0x13      /* interval 2 */
#define      CNR       BASE+0x15      /* convert number */
#define      CR4       BASE+0x17      /* timer/counter set */
#define      CR5       BASE+0x21      /* convert number */
#define      CR6       BASE+0x23      /* ch number */
#define      CR7       BASE+0x25      /* triger e/d & a/d stop */
#define      CR8       BASE+0x27      /* internal triger */
#define      CR9       BASE+0x29      /* fifo reset */
#define      CR10      BASE+0x2b      /* poling status */
#define      DRO       BASE+0x30      /* fifo data */

/*****/
/* eof */
/*****/

```

第 10 章 サンプル・ソフト

10-1 ソフト概要

設定状態の条件にてチャンネル0～7を10 μ s間隔で同時測定を行い各チャンネルのデータをチャンネル番号とA/D変換値に分けてCRT上に表示。

10-2 設定状態

1)ハード設定

項目	初期値	部品番号
アドレスモード	0R (特権及び非特権) / 標準 (A24)	LD7
ベースアドレス	FC4B0000h	LD1-LD6
インターバルタイム	バスケット接続(インターバルタイム1のみ使用)	JP17
チャンネルナンバー	有	DP1-8
FIFOアクセス	ワード	DP1-9
A/Dモード	モード1 (マスタ)	DP1-7
チャンネル 0	ハイゾラ $\pm 10.0[V]$ / 電圧入力	JP1, JP9 / SC7
チャンネル 1	ハイゾラ $\pm 10.0[V]$ / 電圧入力	JP2, JP10 / SC8
チャンネル 2	ハイゾラ $\pm 10.0[V]$ / 電圧入力	JP3, JP11 / SC9
チャンネル 3	ハイゾラ $\pm 10.0[V]$ / 電圧入力	JP4, JP12 / SC10
チャンネル 4	ハイゾラ $\pm 10.0[V]$ / 電圧入力	JP5, JP13 / SC11
チャンネル 5	ハイゾラ $\pm 10.0[V]$ / 電圧入力	JP6, JP14 / SC12
チャンネル 6	ハイゾラ $\pm 10.0[V]$ / 電圧入力	JP7, JP15 / SC13
チャンネル 7	ハイゾラ $\pm 10.0[V]$ / 電圧入力	JP8, JP16 / SC14
SCK IN ターミネット	SC1 : 220 Ω SC2 : 330 Ω	SC1, SC2
SCK OUT ターミネット	SC3 : 220 Ω SC4 : 330 Ω	SC3, SC4
TRG IN ターミネット	SC5 : 220 Ω SC6 : 330 Ω	SC5, SC6

2)ソフト設定

測定チャンネル数 チャンネル0～7 (8チャンネル)
 インターバル時間 10 μ s
 測定回数 2回 (16データ)

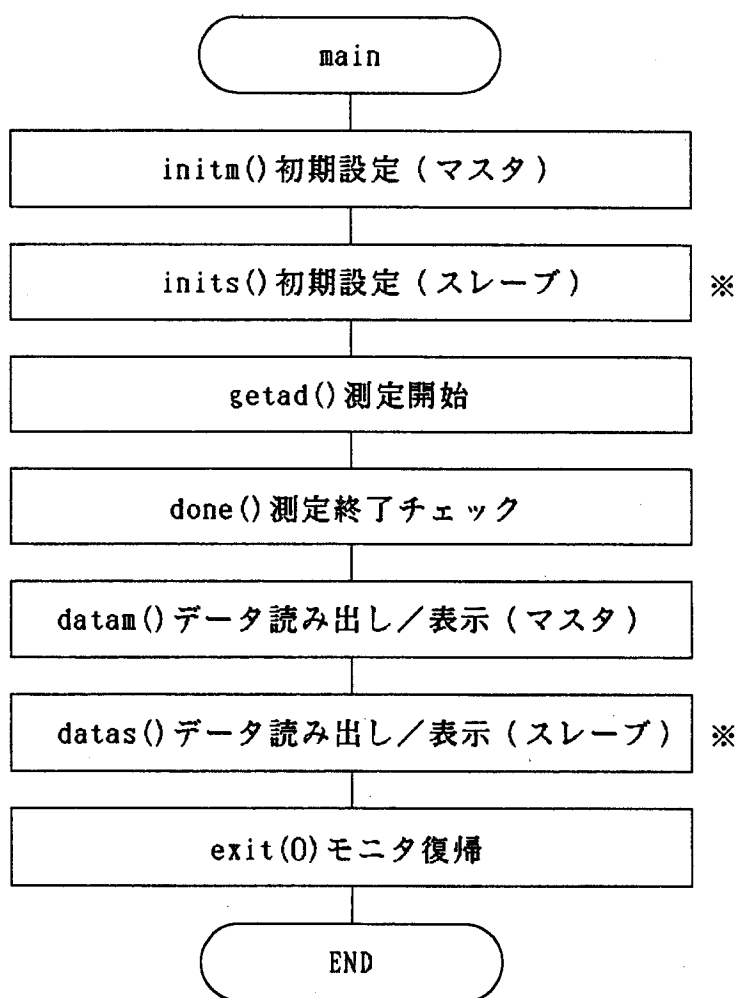
注) 以下に表記するサンプルプログラムは各モード共通に動作することのできるものです。

※印の関数の取扱いを各モードに対応するように実行して下さい。

10-3 プログラム構成

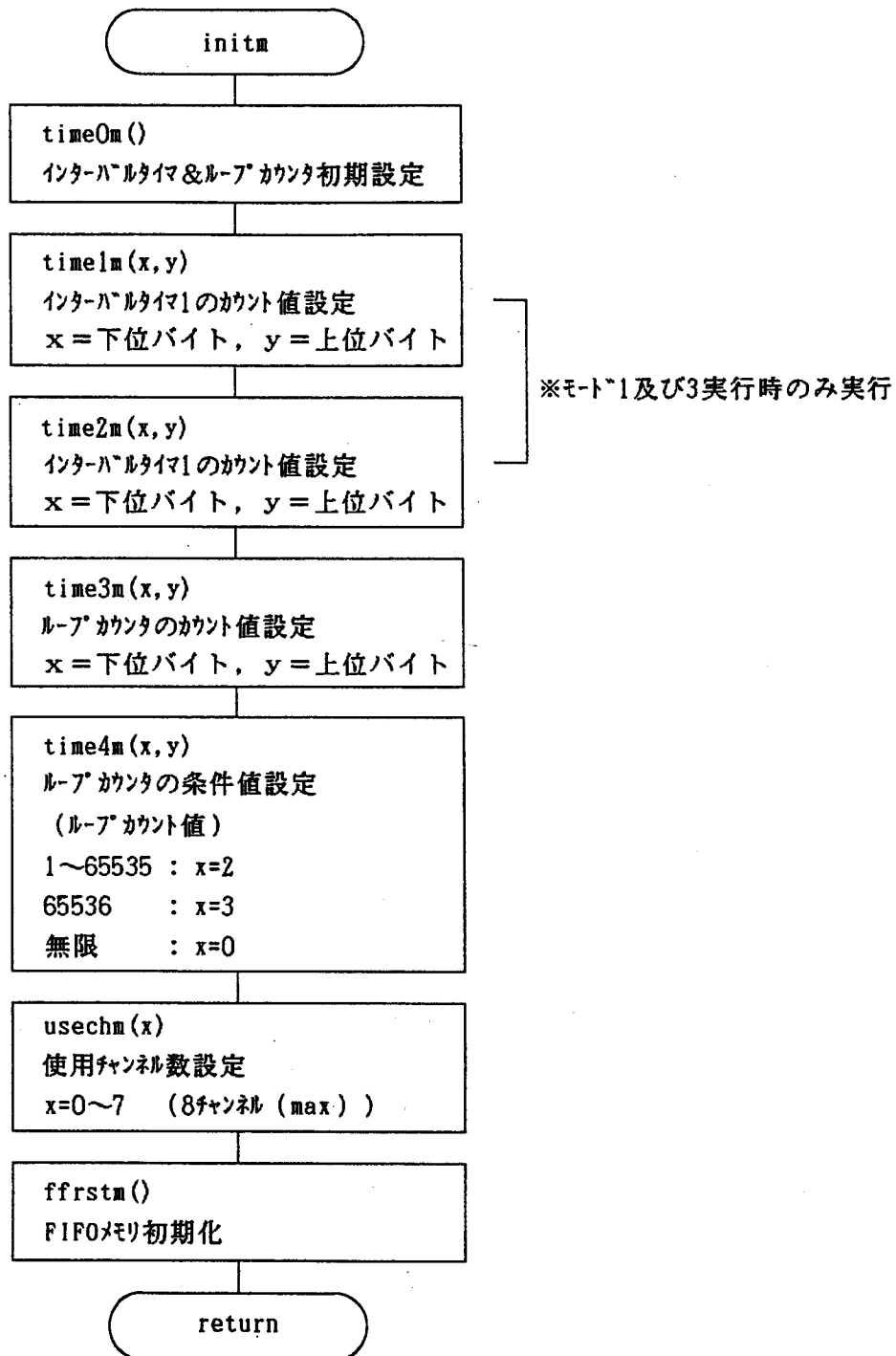
1)関数

①main()

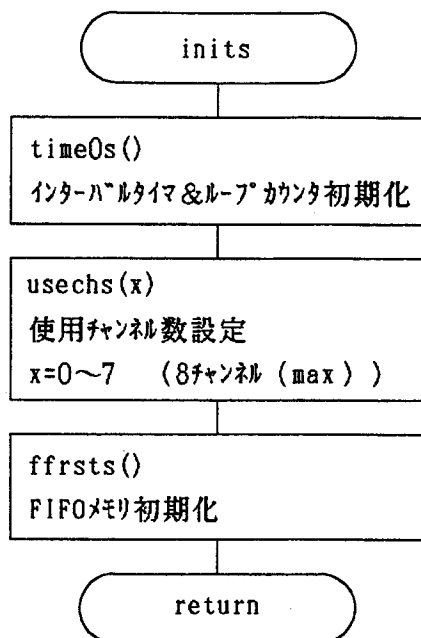


注) ※ : マスタ/スレーブ構成時のみ実行

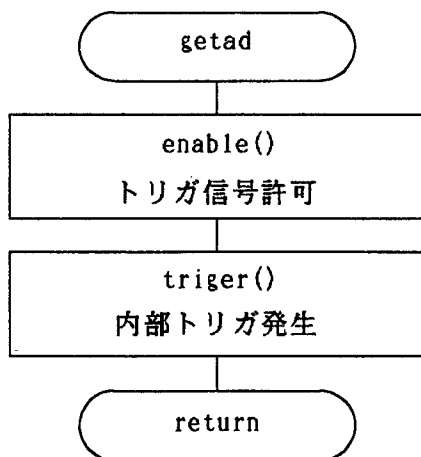
② `initm()`



③ `inits()` ※マスタ/スレーブ構成時のみ実行



④ `getad()`



※モード1及び2設定時のみ 実行

注) モード1 設定時

内部トリガ発生後、インターバルタイムに同期したサンプリングとなる。

モード2 設定時

内部トリガ発生後、外部クロック信号に同期したサンプリングとなる。

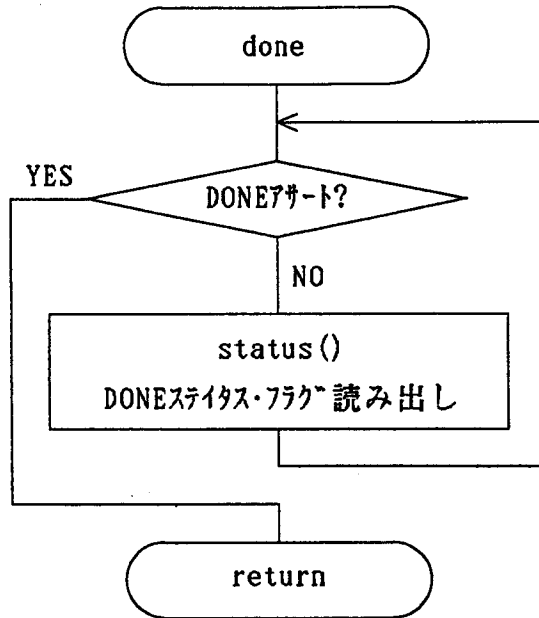
モード3 設定時

外部トリガ入力後、インターバルタイムに同期したサンプリングとなる。

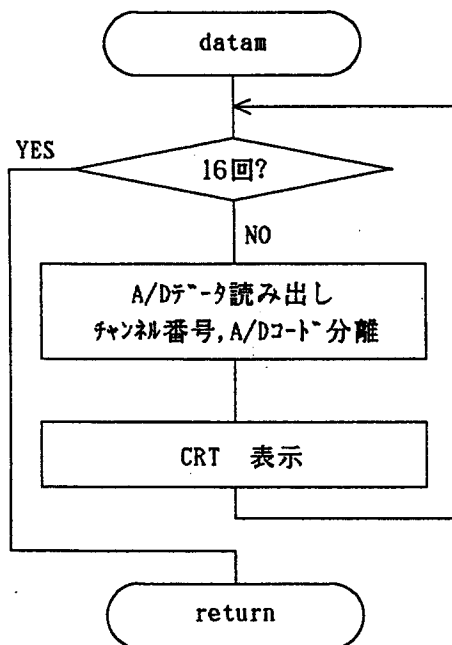
モード4 設定時

外部トリガ入力後、外部クロック信号に同期したサンプリングとなる。

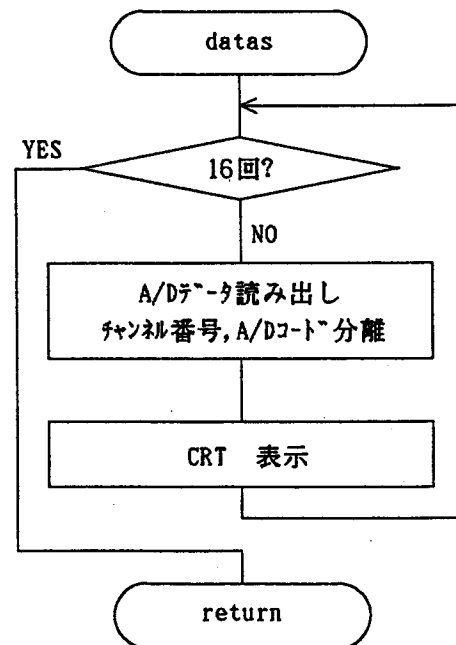
⑤ done ()



⑥ datam ()



⑦ datas () ※マスタ/スレーブ
構成時のみ実行




```

/*****/
/*                                     */
/*   PVME-303 SAMPLE PROGRAM          */
/*                                     */
/*   measur channel: ch0-7            */
/*   intervaltime   : 10 us           */
/*   loop           : 2 (16 a/d data) */
/*                                     */
/*   Written By Internix.inc '93/5/11 */
/*****/

/*-----*/
/* Include  */
/*-----*/

#include <MAST303:H> /* PVME-303 mastre address */
#include <SLAV303:H> /* (master/slave) PVME-303 slave address */
#include <stdio.h>   /* standerd i/o header file */

/*-----*/
/* Main          */
/*-----*/

main()
{
    initm(); /* master initializ */
    /* inits(); /* (master/slave) slave initializ */
    getad(); /* measuer start */
    done(); /* measuer end check */
    datam(); /* master measuer data */
    /* datas(); /* (master/slave) slave measuer data */
    exit(0); /* return to monitor */
}

/*-----*/
/* Initializ master */
/*-----*/

initm()
{
    unsigned char x,y;

    time0m(); /* intervaltimer1,2 & loopcounter initializ */
    x = 0x50; /* (model,3)intervaltimer1:low_byte/word */
    y = 0x00; /* (model,3)intervaltimer1:hi_byte/word */
    time1m(x,y); /* (model,3)intervaltimer1 set */
    x = 0x00; /* (model,3)intervaltimer2:low_byte/word */
    y = 0x00; /* (model,3)intervaltimer2:hi_byte/word */
    time2m(x,y); /* (model,3)intervaltimer2 set */
    x = 0x02; /* loopcounter:low byte/word */
    y = 0x00; /* loopcounter:hi_byte/word */
    time3m(x,y); /* loopcounter set */
    x = 0x02; /* loopcount:1_65535=2/65536=3/mugen=0 */
    time4m(x); /* loopcount set */
    x = 0x07; /* use channel number */
    usechm(x); /* use channel set */
    ffrstm(); /* fifo reset */
}

/*-----*/
/* (master/slave) Initializ slave */
/*-----*/

inits()
{
    unsigned char x,y;

    time0s(); /* intervaltimer1,2 & loopcounter initializ */
    x = 0x07; /* use channel number */
    usechs(x); /* use channel set */
    ffrsts(); /* fifo reset */
}

/*-----*/
/* Measuer start */
/*-----*/

getad()
{
    enable(); /* triger enable */
    triger(); /* (model,2) triger on */
}

```

```

/*-----*/
/* Measuer end */
/*-----*/

done()
{
  unsigned char x;

  for(x = 0x08;x == 0x08;) /* measuer end */
  {
    x = (status() & 0x08); /* read done status */
  }
}

/*-----*/
/* Master data */
/*-----*/

datam()
{
  unsigned short x,y,i;
  unsigned short *dr0;
  dr0 = (unsigned short *) (DR0);
  printf("%n *MASTER MEASUER DATA *");
  for(i = 0;i < 16;i++)
  {
    x = *dr0; /* read a/d data */
    y = (x & 0xf000)>>12; /* channel number */
    x = (x & 0x0fff); /* a/d code */
    printf("%n*CH = %2d:DATA = %3x",y,x);
  }
}

/*-----*/
/* (mstaer/slave) Slave data */
/*-----*/

datas()
{
  unsigned short x,y,i;
  unsigned short *dr0s;
  dr0s = (unsigned short *) (DR0S);
  printf("%n *SLAVE MEASUER DATA *");
  for(i = 0;i < 16;i++)
  {
    x = *dr0s; /* read a/d data */
    y = (x & 0xf000)>>12; /* channel number */
    x = (x & 0x0fff); /* a/d code */
    printf("%n*CH = %2d:DATA = %3x",y,x);
  }
}

/*=====*/
/* Call by initm() */
/*=====*/

time0m()
{
  unsigned char *cr4;
  cr4 = (unsigned char *) (CR4);

  *cr4 = 0x34; /* intervaltimer1 set */
  *cr4 = 0x74; /* intervaltimer2 set */
  *cr4 = 0xba; /* loopcounter set */
}

time1m(x,y)
  unsigned char x,y;
{
  unsigned char *tr0;
  tr0 = (unsigned char *) (TR0);
  *tr0 = x; /* intervaltimer1 low byte/word */
  *tr0 = y; /* intervaltimer1 hi byte/word */
}

time2m(x,y)
  unsigned char x,y;
{
  unsigned char *tr1;
  tr1 = (unsigned char *) (TR1);
  *tr1 = x; /* intervaltimer2 low byte/word */
  *tr1 = y; /* intervaltimer2 hi byte/word */
}

time3m(x,y)
  unsigned char x,y;

```

```

{
  unsigned char *cnr;
  cnr = (unsigned char *) (CNR);
  *cnr = x; /* loopcounter low byte/word */
  *cnr = y; /* loopcounter hi byte/word */
}
time4m(x)
  unsigned char x;
  {
    unsigned char *cr5;
    cr5 = (unsigned char *) (CR5);
    *cr5 = x; /* loopcount */
  }
usechm(x)
  unsigned char x;
  {
    unsigned char *cr6;
    cr6 = (unsigned char *) (CR6);
    *cr6 = x; /* use channel */
  }
ffirstm()
  {
    unsigned char *cr9;
    cr9 = (unsigned char *) (CR9);
    *cr9 = 0x00; /* fifo reset */
  }

/*=====*/
/* (msater/slave) Call by inits() */
/*=====*/
time0s()
  {
    unsigned char *cr4s;
    cr4s = (unsigned char *) (CR4S);
    *cr4s = 0x34; /* intervaltmer1 set */
    *cr4s = 0x74; /* intervaltmer2 set */
    *cr4s = 0xba; /* loopcounter set */
  }
usechs(x)
  unsigned char x;
  {
    unsigned char *cr6s;
    cr6s = (unsigned char *) (CR6);
    *cr6s = x; /* use channel number */
  }
ffirsts()
  {
    unsigned char *cr9s;
    cr9s = (unsigned char *) (CR9S);
    *cr9s = 0x00; /* fifo reset */
  }

/*=====*/
/* Call by getad() */
/*=====*/
enable()
  {
    unsigned char *cr7;
    cr7 = (unsigned char *) (CR7);
    *cr7 = 0x01; /* triger enable */
  }
triger()
  {
    unsigned char *cr8;
    cr8 = (unsigned char *) (CR8);
    *cr8 = 0xff; /* (model,2) triger on */
  }

/*=====*/
/* Call by done() */
/*=====*/
status()
  {
    unsigned char *cr10;
    unsigned char x;
    cr10 = (unsigned char *) (CR10);
    x = (*cr10 & 0x0f);
    return(x);
  }
/*=====*/
/* e.o.f */
/*=====*/

```

```

/*****/
/*
/*          PVME-303 SAMPLE PROGRAM (MASTER ADDRESS)          */
/*
/*
/*****/

#define      BASE      0xfc4b0000          /* base adrs */
#define      CR0       BASE+0x01          /* irq ef */
#define      CR1       BASE+0x03          /* irq hf */
#define      CR2       BASE+0x05          /* irq ff */
#define      CR3       BASE+0x07          /* irq done */
#define      VR0       BASE+0x09          /* vct ef */
#define      VR1       BASE+0x0b          /* vct hf */
#define      VR2       BASE+0x0d          /* vct ff */
#define      VR3       BASE+0x0f          /* vct done */
#define      TR0       BASE+0x11          /* interval 1 */
#define      TR1       BASE+0x13          /* interval 2 */
#define      CNR       BASE+0x15          /* convert number */
#define      CR4       BASE+0x17          /* timer/counter set */
#define      CR5       BASE+0x21          /* convert number */
#define      CR6       BASE+0x23          /* ch number */
#define      CR7       BASE+0x25          /* trigger e/d & a/d stop */
#define      CR8       BASE+0x27          /* internal trigger */
#define      CR9       BASE+0x29          /* fifo reset */
#define      CR10      BASE+0x2b          /* poling status */
#define      DR0       BASE+0x30          /* fifo data */

/*****/
/* eof */
/*****/

```

```

/*****/
/*
/*          PVME-303 SAMPLE PROGRAM (SLAVE ADDRESS)          */
/*
/*
/*****/

#define      BASES     0xfc4b0000          /* base adrs */
#define      CR0S      BASES+0x10001      /* irq ef */
#define      CR1S      BASES+0x10003      /* irq hf */
#define      CR2S      BASES+0x10005      /* irq ff */
#define      CR3S      BASES+0x10007      /* irq done */
#define      VR0S      BASES+0x10009      /* vct ef */
#define      VR1S      BASES+0x1000b      /* vct hf */
#define      VR2S      BASES+0x1000d      /* vct ff */
#define      VR3S      BASES+0x1000f      /* vct done */
#define      TR0S      BASES+0x10011      /* interval 1 */
#define      TR1S      BASES+0x10013      /* interval 2 */
#define      CNRS      BASES+0x10015      /* convert number */
#define      CR4S      BASES+0x10017      /* timer/counter set */
#define      CR5S      BASES+0x10021      /* convert number */
#define      CR6S      BASES+0x10023      /* ch number */
#define      CR7S      BASES+0x10025      /* trigger e/d & a/d stop */
#define      CR8S      BASES+0x10027      /* internal trigger */
#define      CR9S      BASES+0x10029      /* fifo reset */
#define      CR10S     BASES+0x1002b      /* poling status */
#define      DROS      BASES+0x10030      /* fifo data */

/*****/
/* eof */
/*****/

```

第 1 1 章 そ の 他

10-1 保証について

添付の保証返信カードに御記入の上、御返送下さい。

本ボードの無償保証期間は、お買い求めの日から1年です。ただし、環境条件、操作方法、その他正常な使用方法で使用されるものに限りです。

下記に該当するものについては、無償保証期間であっても無償修理対象とはなりません。

- a. お客様の誤った使用方法、あるいは不注意によって生じた故障、損傷。
- b. 不当な改造、修理に帰すと認められる故障、損傷。
- c. 自然災害によって生じた故障、損傷。
- d. 保証書の御提示がない場合。

10-2 オーダーリング・インフォメーション

PVME-303

1993年 6月 M303-01 (第一版)

- ・本マニュアルの内容については、予告なく変更修正する場合があります。
- ・本マニュアルに記載されている以外のご使用によって損害が発生した場合、当社では責任を負いかねますのでお取扱いには十分ご注意下さい。

インターニックス株式会社
八王子開発センター

〒192 東京都八王子市片倉町59-10

TEL. 0426 (48) 5200

FAX. 0426 (48) 5201