2.18 4チャネルA/D変換による電圧測定

4チャネルA/D変換による電圧測定	使用機能	A/Dコンバータ

仕様

(1) A/Dコンバータを使用して、4チャネルのA/D変換による電圧測定を行ないます。
(2) 図1に示すように、4チャネルの電圧をH8/3664に入力し、A/D変換した結果をRAMに格納します。

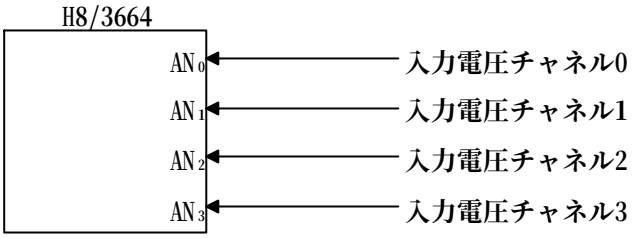


図1　4チャネルA/D変換による電圧測定

使用機能説明

(1) 本タスク例では、A/Dコンバータを使用して、4チャネルA/D変換による電圧測定を行ないます。
(a) 図2にA/Dコンバータのブロック図を示します。以下にA/Dコンバータのブロック図について説明します。
- 本タスク例では、A/D変換スピードを12.4μsに設定しています。
- A/D データレジスタ（ADDRA～D）は、A/D 変換結果を格納するための16ビットのリード専用レジスタで、ADDRA～ADDRDの4本あります。10ビットの変換データはA/Dデータレジスタのビット15からビット6に格納されます。下位6ビットの読み出し値は常に0です。CPUとの間のデータバスは8ビット幅で、上位バイトはCPUから直接リードできますが、下位バイトは上位バイトリード時にテンポラリレジスタに転送されたデータが読み出されます。このためA/Dデータレジスタをリードする場合は、ワードアクセスするか上位バイトのみリードしてください。ADDRの初期値はH'0000 です。
- A/Dコントロール/ステータスレジスタ（ADCSR）は、A/D変換器の制御ビットと変換終了ステータスビットで構成されています。
- アナログ入力端子0～7（AN_0～AN_7）は、入力電圧チャネル0～7の入力端子です。
- アナログ電源（AV_{CC}）は、アナログ部の電源および基準電圧端子です。
- アナロググランド（AV_{SS}）は、アナログ部のグランドおよび基準電圧端子です。
- 本タスク例では、アナログ入力端子0～3（AN_0～AN_3）を使用し、4チャネルA/D変換による電圧測定を行ないます。

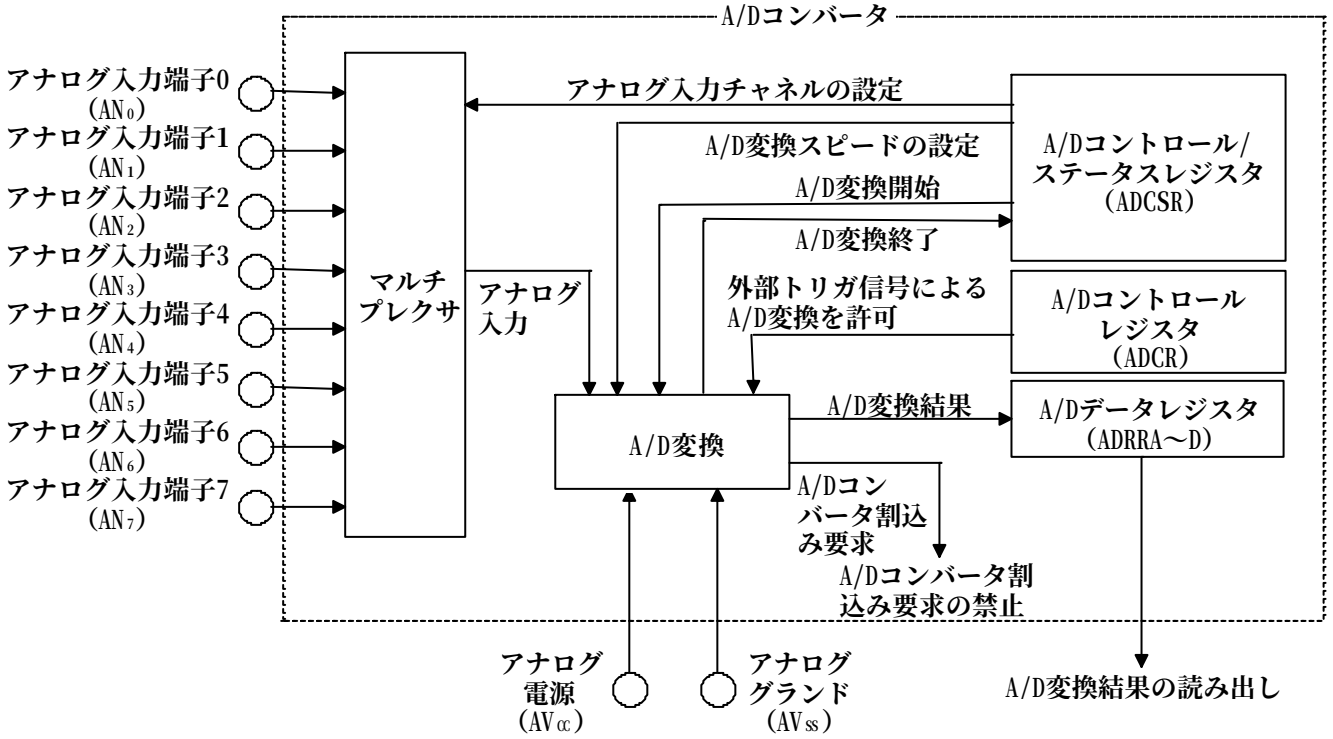


図2　A/Dコンバータのブロック図

4チャネルA/D変換による電圧測定	使用機能	A/Dコンバータ

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、4チャネルA/D変換による電圧測定を行ないます。

表1　機能割付け

機能	機能割付け
ADCSR	A/D変換の開始、終了、ステータス、スピードの設定、およびアナログ入力端子の指定
ADDRA～D	A/D変換された結果を格納
AN_0～AN_7	入力電圧チャネル0～7の入力端子（但し、本タスク例では、AN_0～AN_3を使用）
AV_{CC}	アナログ部の電源および基準電圧端子
AV_{SS}	アナログ部のグランドおよび基準電圧端子

動作説明

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理により4チャネルA/D変換による電圧測定を行ないます。

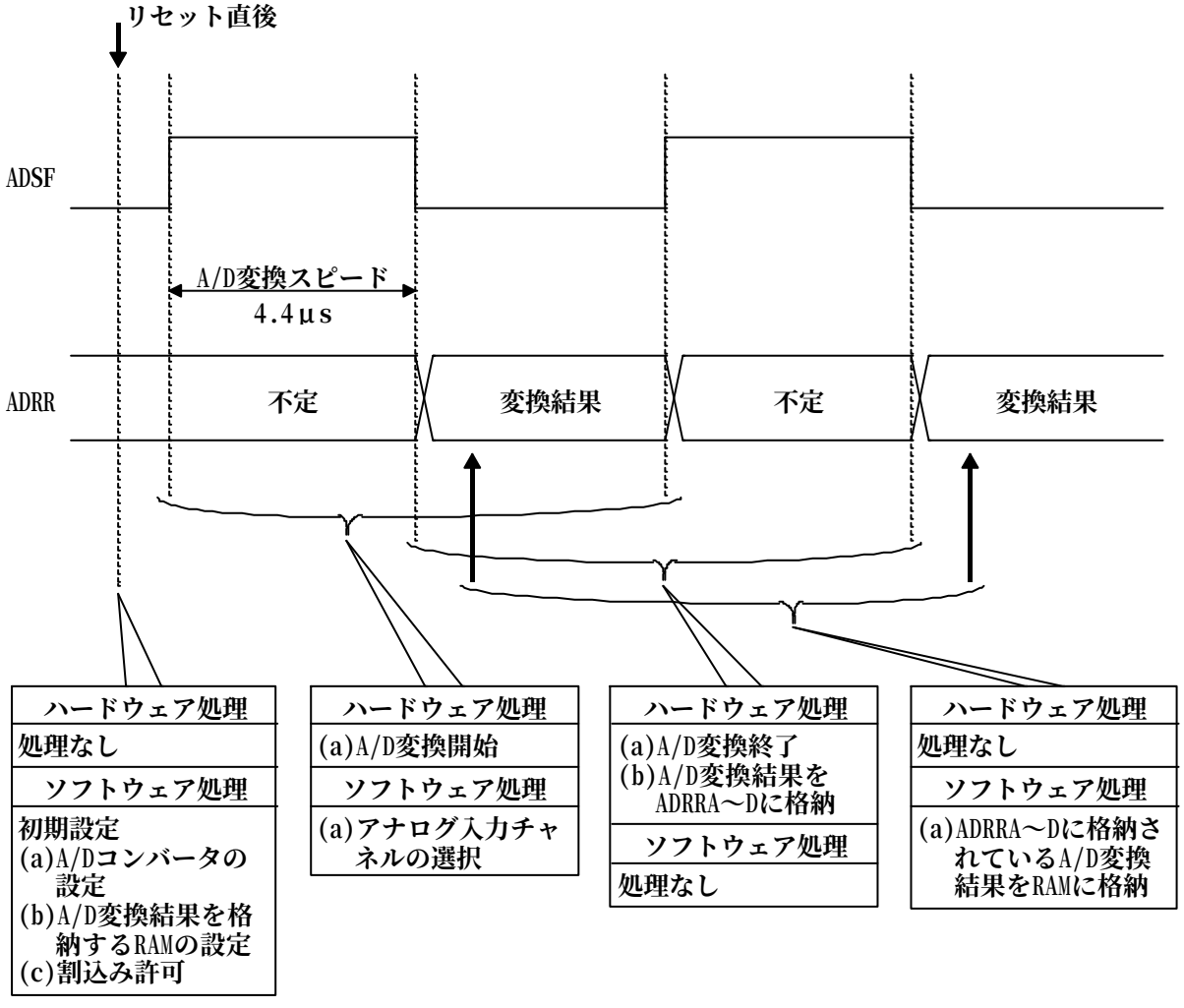


図3　4チャネルA/D変換による電圧測定の動作原理

4チャネルA/D変換による電圧測定	使用機能	A/Dコンバータ

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2　モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	A/Dコンバータの設定、割込みの許可、アナログ入力チャネルの選択、A/D変換の開始、A/D変換終了後ADRRA～Dに格納されたA/D変換結果をRAMに格納、アナログ入力チャネル0～3までのA/D変換を行なうと終了

(2) 引数の説明

表3に本タスク例における引数の説明を示します。

表3　引数の説明

引数名	機能	使用モジュール名	データ長	入出力
ADRRA	アナログ入力チャネル0のA/D変換結果を格納	メインルーチン	2バイト	出力
ADRRB	アナログ入力チャネル1のA/D変換結果を格納	メインルーチン	2バイト	出力
ADRRC	アナログ入力チャネル2のA/D変換結果を格納	メインルーチン	2バイト	出力
ADRRD	アナログ入力チャネル3のA/D変換結果を格納	メインルーチン	2バイト	出力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4　使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
ADCSR	ADF	A/Dコントロール/ステータスレジスタ（A/Dエンドフラグ） ：ADF="1"のとき、単一モードでA/D変換が終了	H'FFB8 ビット7	0
	ADIE	A/Dコントロール/ステータスレジスタ（A/Dインタラプトイネーブル） ：ADIE="1"のとき、ADFによるA/D変換終了割込み要求（ADI） がイネーブルになります。	H'FFB8 ビット6	0
	ADST	A/Dコントロール/ステータスレジスタ（A/Dスタート） ：このビットを１にセットするとA/D 変換を開始します。単一 モードではA/D変換を終了すると自動的にクリアされます。	H'FFB8 ビット5	0
	SCAN	A/Dコントロール/ステータスレジスタ（スキャンモード） ：SCAN="0"のとき、単一モードを選択 ：SCAN="1"のとき、スキャンモードを選択	H'FFB8 ビット4	0
	CKS	A/Dコントロール/ステータスレジスタ（クロックセレクト） ：SCAN="0"のとき、A/D変換時間を134ステート(max)に設定 ：SCAN="1"のとき、A/D変換時間を70ステート(max)に設定 ※変換時間の切換えは、ADST=0の状態で行ってください。	H'FFB8 ビット3	0
	CH2 CH1 CH0	A/Dコントロール/ステータスレジスタ（チャンネルセレクト2～0） ：CH2="0"、CH1="0"、CH0="0"のとき、AN_0を選択 ：CH2="0"、CH1="0"、CH0="1"のとき、AN_1を選択 ：CH2="0"、CH1="1"、CH0="0"のとき、AN_2を選択 ：CH2="0"、CH1="1"、CH0="1"のとき、AN_3を選択	H'FFB8 ビット2 ビット1 ビット0	0 CH2="0" CH1="0" CH0="0"
ADRRA		A/Dデータレジスタ A：A/D変換結果の16ビットデータを格納	H'FFB0	H'0000
ADRRB		A/Dデータレジスタ B：A/D変換結果の16ビットデータを格納	H'FFB2	H'0000
ADRRC		A/Dデータレジスタ C：A/D変換結果の16ビットデータを格納	H'FFB4	H'0000
ADRRD		A/Dデータレジスタ D：A/D変換結果の16ビットデータを格納	H'FFB6	H'0000

(4) 使用RAM説明

表4に本タスク例における使用RAM説明を示します。

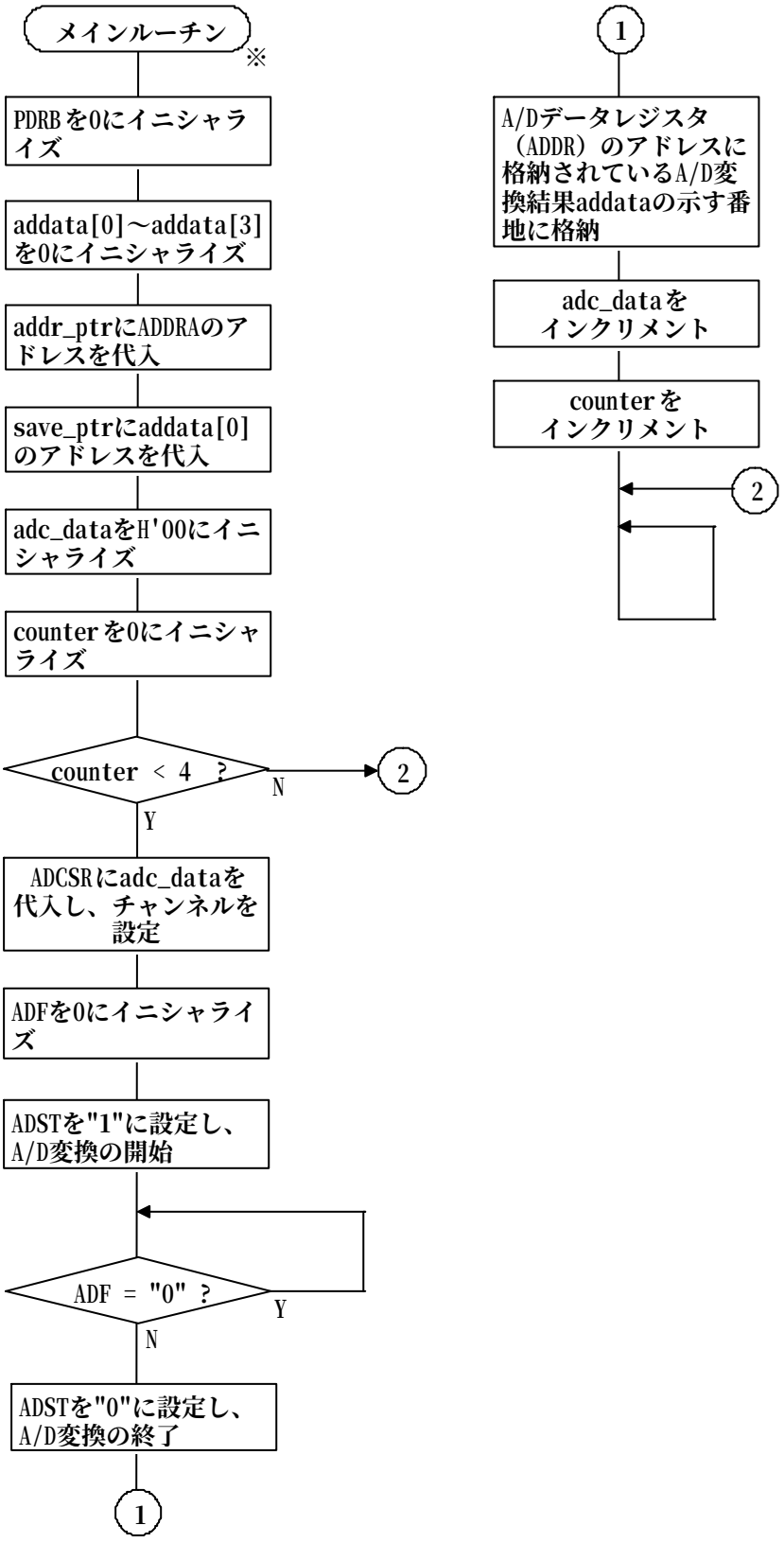
表4　使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
addata[4]	RAMに格納する為のデータの変数	H'FB80	メインルーチン
counter	4チャンネルA/D変換の回数をカウント	H'FB88	メインルーチン

4チャネルA/D変換による電圧測定	使用機能	A/Dコンバータ

フローチャート

(a) メインルーチン



※本例ではスタックポインタはINIT.SRC（アセンブリ言語）で設定してあります。

4チャネルA/D変換による電圧測定	使用機能	A/Dコンバータ

プログラムリスト

INIT.SRC（プログラムリスト）

```
        .EXPORT  _INIT
        .IMPORT  _main
;
        .SECTION    P,CODE
_INIT:
        MOV.W    #H'FF80,R7
        LDC.B    #B'10000000,CCR
        JMP              @_main
;
        .END
```

```
/************************************************************/
/*                                                          */
/*  H8/300H Tiny Series -H8/3664-                           */
/*  Application Note                                        */
/*                                                          */
/*  'Voltage Measurement by 4-Channel A/D                   */
/*   Converter'                                             */
/*                                                          */
/*  Function                                                */
/*  : A/D Converter                                         */
/*                                                          */
/*  External Clock : 16MHz                                  */
/*  Internal Clock : 16MHz                                  */
/*  Sub Clock      : 32.768kHz                              */
/*                                                          */
/************************************************************/

#include    <machine.h>

/************************************************************/
/*  Symbol Defnition                                        */
/************************************************************/
struct BIT {
    unsigned char   b7:1;     /* bit7 */
    unsigned char   b6:1;     /* bit6 */
    unsigned char   b5:1;     /* bit5 */
    unsigned char   b4:1;     /* bit4 */
    unsigned char   b3:1;     /* bit3 */
    unsigned char   b2:1;     /* bit2 */
    unsigned char   b1:1;     /* bit1 */
    unsigned char   b0:1;     /* bit0 */
};
#define   TMA        *(volatile unsigned char *)0xFFA6 /* Timer Mode Register A          */
#define   TCA        *(volatile unsigned char *)0xFFA7 /* Timer Counter A                */
#define   PDR8       *(volatile unsigned char *)0xFFDB /* Port Data Register 8           */
#define   P81        PDR8_BIT.b1                       /* Port Data Register 8 bit1      */
#define   PCR8       *(volatile unsigned char *)0xFFEB /* Port Control Register 8        */
#define   PCR81      PCR8_BIT.b1                       /* Port Control Register 8  bit1  */
#define   IENR1_BIT  (*(struct BIT *)0xFFF4)           /* Interrupt Enable Register 1    */
#define   IENTA      IENR1_BIT.b6                      /* Timer A Interrupt Enable       */
#define   IRR1_BIT   (*(struct BIT *)0xFFF6)           /* Interrupt Request Register 1   */
#define   IRRTA      IRR1_BIT.b6                       /* Timer A Interrupt Request Flag */
#define   ADDRA      *(volatile unsigned int *)0xFFB0  /* A/D Data Register A            */
#define   ADDRB      *(volatile unsigned int *)0xFFB2  /* A/D Data Register B            */
#define   ADDRC      *(volatile unsigned int *)0xFFB4  /* A/D Data Register C            */
#define   ADDRD      *(volatile unsigned int *)0xFFB6  /* A/D Data Register D            */
#define   ADCSR      *(volatile unsigned char *)0xFFB8 /* A/D Control/Status Register    */
#define   ADCSR_BIT  (*(struct BIT *)0xFFB8)           /* A/D Control/Status Register    */
#define   ADF        ADCSR_BIT.b7                      /* A/D END Flag                   */
#define   ADIE       ADCSR_BIT.b6                      /* A/D Interrupt Enable           */
#define   ADST       ADCSR_BIT.b5                      /* A/D Start                      */
#define   SCAN       ADCSR_BIT.b4                      /* A/D Scan Mode                  */
#define   CKS        ADCSR_BIT.b3                      /* A/D Clock Select               */
#define   CH2        ADCSR_BIT.b2                      /* Channel Select 2               */
#define   CH1        ADCSR_BIT.b1                      /* Channel Select 1               */
#define   CH0        ADCSR_BIT.b0                      /* Channel Select 0               */
#define   PDRB       *(volatile unsigned char *)0xFFDD /* Port Data Register B           */
```

4チャネルA/D変換による電圧測定	使用機能	A/Dコンバータ

プログラムリスト

```
/************************************************************/
/*  関数定義                                                */
/************************************************************/
extern  void    INIT( void );                   /* SP Set                                   */
void    main    ( void );
/************************************************************/
/*  RAM define                                              */
/************************************************************/
    unsigned int    addata[4];
    unsigned char   counter;
/************************************************************/
/*  Vector Address                                          */
/************************************************************/
#pragma section     V1                          /* VECTOR SECTOIN SET                       */
void (*const VEC_TBL1[])(void) = {
/* 0x00 - 0x0f */
    INIT                                        /* 00 Reset                                 */
};
#pragma section                                 /* P                                        */
/************************** ***********************************/
/*  Main Program                                            */
/************************************************************/
void main ( void )
{
    unsigned int    *addr_ptr,*save_ptr;
    unsigned char   adc_data;
    unsigned int    cnt;

    PDRB = 0;                                   /* Clear PDRB                               */

    addata[0] = 0;                              /* Clear adddata[0]                         */
    addata[1] = 0;                              /* Clear adddata[1]                         */
    addata[2] = 0;                              /* Clear adddata[2]                         */
    addata[3] = 0;                              /* Clear adddata[3]                         */

    addr_ptr = &ADDRA;
    save_ptr = &addata[0];
    adc_data = 0x00;                            /* Clear adc_data                           */
    counter = 0;                                /* Clear counter                            */
    while( counter < 4 ){                       /* A/D Convert END ?                        */
        ADCSR = adc_data;                       /* Select A/D Convert Time & Analog Input Channel */
        ADF = 0;                                /* Initialize ADF                           */
        ADST = 1;                               /* Start A/D Convert                        */

        while(ADF == 0){                        /* A/D Convert End ?                        */
            ;
        }

        ADST = 0;                               /* Stop A/D Convert                         */

        *(save_ptr + counter) = *(addr_ptr + counter);

        adc_data++;
        counter++;                              /* Decrement A/D Convert Counter            */
    }

    while(1){
       ;
    }
}
```

リンクアドレス指定

セクション名	アドレス
CV1	H'0000
P	H'0100
B	H'FB80