To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

H8/300H Tiny Series

Measuring Voltages by 4-Channel A/D Conversion

Introduction

The A/D converter is used to measure the four voltages by 4-channel A/D conversion.

Target Device

H8/3664

Contents

1. Specifications

- The A/D converter is used to measure the four voltages by 4-channel A/D conversion.
- Inputs voltage to the H8/3664 Series through four channels, and stores the A/D conversion result in RAM, as shown in figure 1.

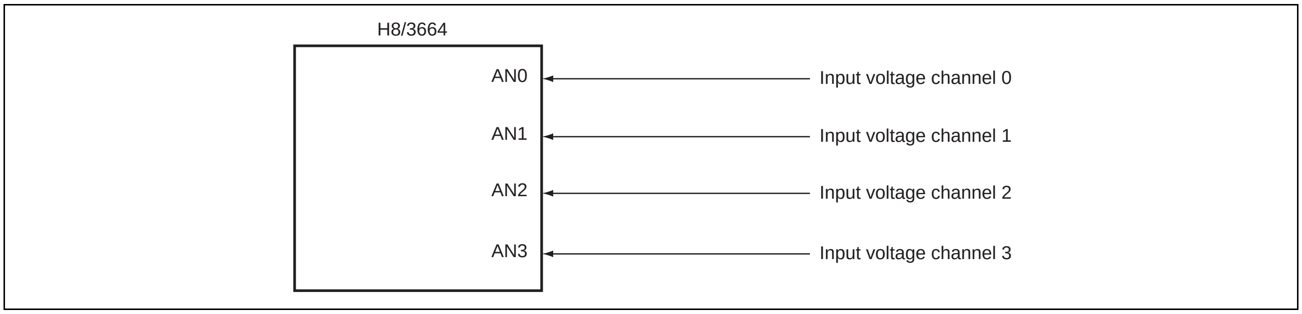


Figure 1 Voltage Measurement by 4-Channel A/D Conversion

2. Description of Functions Used

In this sample task, the voltages are measured using the A/D converter for 4-channel A/D conversion.

Figure 2 is a block diagram of the A/D converter. The elements of the block diagram are described below.

- In this sample task, the A/D conversion time is set to 12.4 µs per channel.
- The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only ADDR registers used to store the results of A/D conversion. The converted 10-bit data is stored in bits 15 to 6 of the respective ADDR. The lower 6 bits are always read as 0. The data bus between the CPU and the A/D converter is 8 bits wide. The upper byte can be read directly from the CPU, however the lower byte should be read via a temporary register. The temporary register contents are transferred from the ADDR when the upper byte of data is read. When reading from ADDR, read the upper byte only or read in word units. Each ADDR is initialized to H'0000.
- The A/D control/status register (ADCSR) contains the control bits and conversion end status bits of the A/D converter.
- Analog input pins 0 to 7 (AN0 to AN7) are input pins for input voltage channels 0 to 7.
- The analog power supply (AV_{CC}) is a power supply and reference voltage pin for the analog block.
- The analog ground (AV_{SS}) is a ground and reference voltage pin for the analog block.
- In this sample task, the voltages of analog input pins 0 to 3 (AN0 to AN3) are measured by 4-channel A/D conversion.

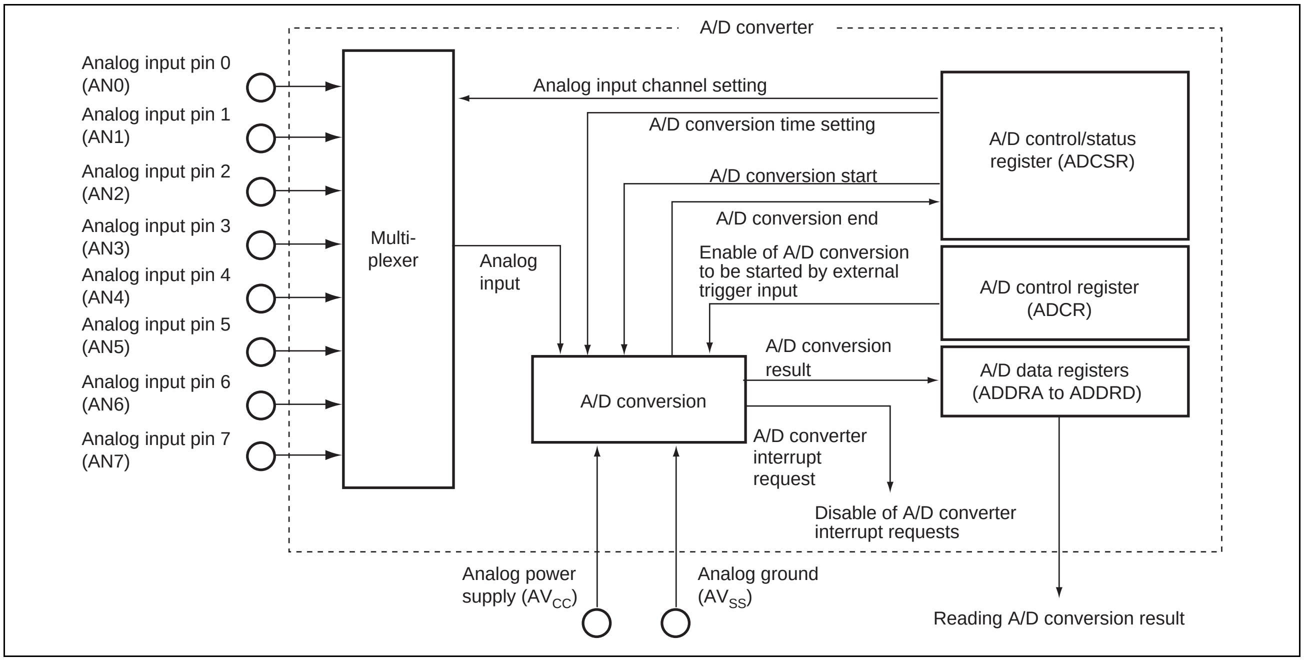


Figure 2 A/D Converter

Table 1 lists the function allocation for this sample task. The functions listed in table 1 are allocated for measuring voltages by 4-channel A/D conversion.

Table 1 Function Allocation

Function	Description
ADCSR	Sets start, end, status, and conversion time of A/D conversion, and selects analog input pins
ADDRA to ADDRD	Stores the A/D conversion result
AN0 to AN7	Input pins for input voltage channels 0 to 7 (in this sample task, only AN0 to AN3 are used)
AV_{CC}	Power supply and reference voltage pin for the analog block
AV_{SS}	Ground and reference voltage pin for the analog block

3. Description of Operations

Figure 3 shows this sample task's principle of operation. The hardware and software processing shown in figure 3 applies 4-channel A/D conversion to measure voltages.

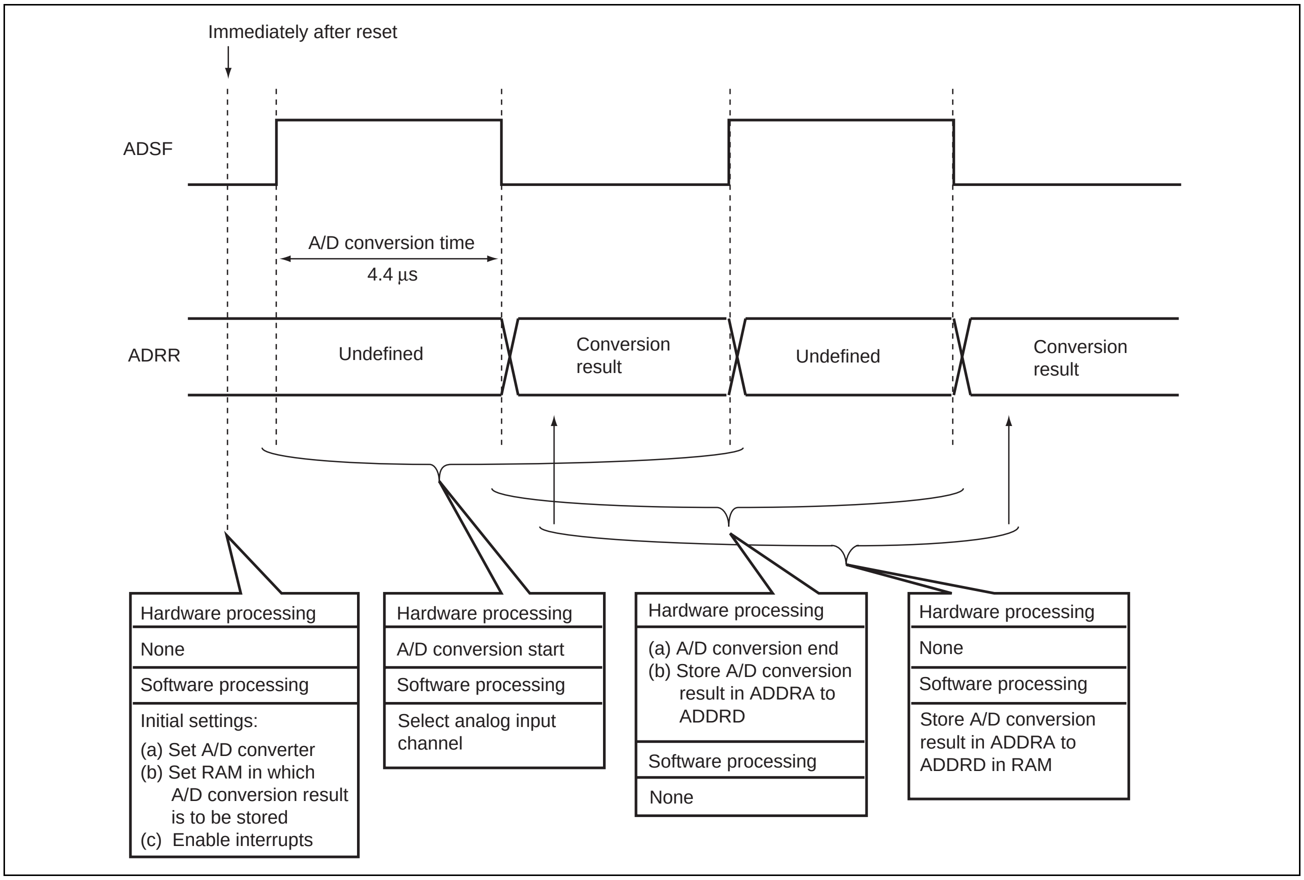


Figure 3 Operation Principle: Voltage Measurement by 4-Channel A/D Conversion

4. Description of Software

4.1 Description of Module

Table 2 describes the software used in this sample task.

Table 2 Description of Modules

Module Name	Label Name	Function
Main routine	main	Sets the A/D converter, enables interrupts, selects the analog input channel, starts A/D conversion, moves the A/D conversion result stored in ADDRA to ADDRD to RAM after A/D conversion ends, and the A/D converter stops after performing A/D conversion by analog input channels 0 to 3.

4.2 Description of Arguments

Table 3 describes the arguments used in this sample task.

Table 3 Description of Arguments

Argument Names	Function	Used in	Data Length	I/O
ADDRA	Stores the A/D conversion result of analog input channel 0.	Main routine	2 bytes	Output
ADDRB	Stores the A/D conversion result of analog input channel 1.	Main routine	2 bytes	Output
ADDRC	Stores the A/D conversion result of analog input channel 2.	Main routine	2 bytes	Output
ADDRD	Stores the A/D conversion result of analog input channel 3.	Main routine	2 bytes	Output

4.3 Description of Internal Registers

Table 4 describes the internal registers used in this sample task.

Table 4 Description of Internal Registers

Register Name		Function	Address	Setting
ADCSR	ADF	A/D control/status register (A/D end flag): [Setting conditions] - Completion of A/D conversion in single mode - Completion of one round of conversion for all selected channels in scan mode. [Clearing condition] - Writing of 0 to the bit after having read it as 1	H'FFB8 Bit 7	0
	ADIE	A/D control/status register (A/D interrupt enable): When this bit is set to 1, A/D conversion end interrupt (ADI) requests from ADF are enabled.	H'FFB8 Bit 6	0
	ADST	A/D control/status register (A/D start): When ADST is set to 1, A/D conversion starts. When A/D conversion ends in single mode, ADST is automatically cleared. In scan mode, the sequence of A/D conversion for the selected channels is repeatedly performed until this bit is cleared by software, resetting, or entry to standby mode.	H'FFB8 Bit 5	0

Table 4 Description of Internal Registers (cont)

Register Name		Function	Address	Setting
ADCSR (cont)	SCAN	A/D control/status register (scan mode): Selection of the A/D conversion mode. 0: Single mode 1: Scan mode	H'FFB8 Bit 4	0
	CKS	A/D control/status register (clock select): Setting for A/D conversion time. 0: A/D conversion time = 134 states (max.) 1: A/D conversion time = 70 states (max.) Note: Clear the ADST bit to 0 before switching the conversion time.	H'FFB8 Bit 3	0
	CH2 CH1 CH0	A/D control/status register (channel select 2 to 0): When CH2, CH1, and CH0 are all cleared to 0, AN0 is selected. When CH2 and CH1 are both cleared to 0 and CH2 is set to 1, AN1 is selected. When CH2 and CH0 are both cleared to 0 and CH1 is set to 1, AN2 is selected. When CH2 is cleared to 0 and CH1 and CH0 are both set to 1, AN3 is selected.	H'FFB8 Bit 2 Bit 1 Bit 0	CH2 = 0 CH1 = 0 CH0 = 0
ADDRA		A/D data register A: Stores the 16-bit data of the A/D conversion result.	H'FFB0	H'0000
ADDRB		A/D data register B: Stores the 16-bit data of the A/D conversion result.	H'FFB2	H'0000
ADDRC		A/D data register C: Stores the 16-bit data of the A/D conversion result.	H'FFB4	H'0000
ADDRD		A/D data register D: Stores the 16-bit data of the A/D conversion result.	H'FFB6	H'0000

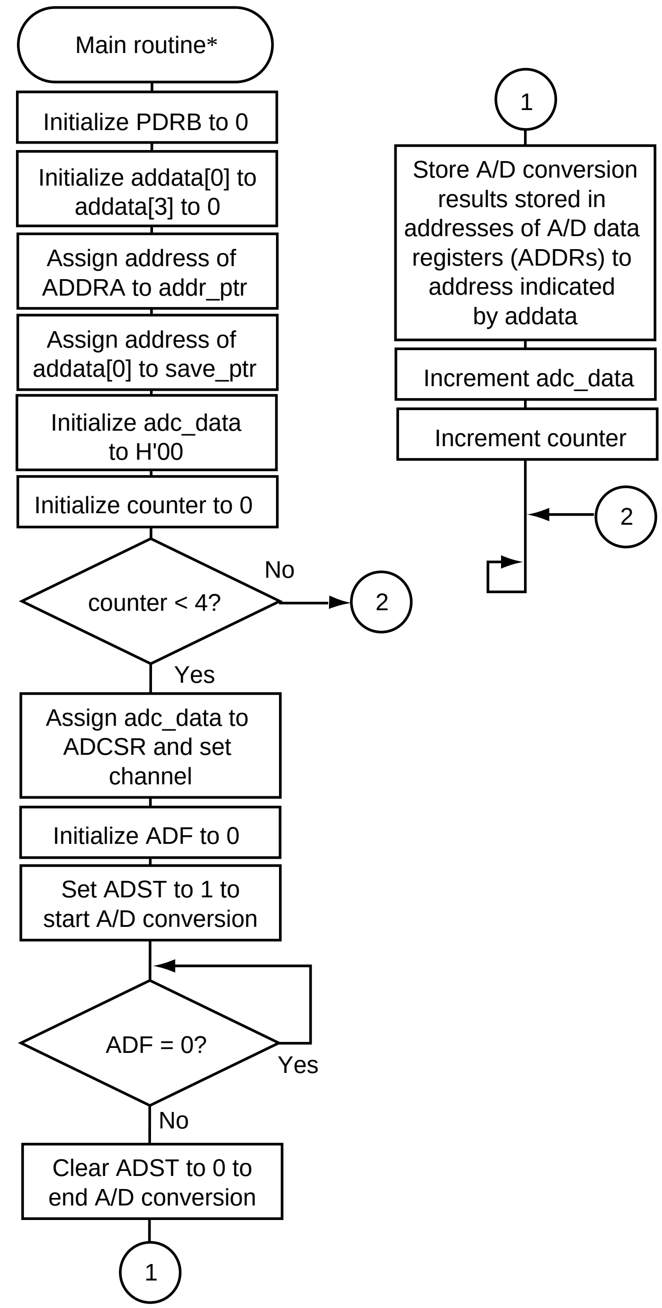
4.4 Description of RAM

Table 5 describes the RAM used in this sample task.

Table 5 Description of RAM

Label Name	Function	Address	Used in
addata[4]	Data variables for RAM storage	H'FB80	Main routine
counter	Counts number of times of 4-channel A/D conversion	H'FB88	Main routine

5. Flowchart



Note: * In this sample task, the stack pointer is set in INIT.SRC (assembly).

Figure 4 Flowchart for Main Routine

6. Program Listing

INIT.SRC (Program listing)

```
      .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           */
```

```
/************************************************************/
/*     Function Definition                                  */
/************************************************************/
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){
        ;
    }
}
```

Link Address Setting:

Section Name	Address
CV1	H'0000
P	H'0100
B	H'FB80

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

Rev.	Date	Description: Page	Description: Summary
2.00	Sep.01.06	All pages	Format has been changed from Hitachi version to Renesas version.