Hitachi America, Ltd.

Application Note

H8/330

Cristian Tomescu

A/D Conversion Using the H8/330

INTRODUCTION

Some of the H8/300 family microcontrollers have a built-in analog-to-digital (A/D) peripheral module, in addition to the more common ROM, RAM, 8- and 16-bit timers, and SCI modules. Specifically, the H8/330, H8/338 family, and the H8/329 family offer an 8 channel input A/D converter with an 8-bit resolution, while the H8/350 extends it to a 16 channel input capability.

DESCRIPTION OF OPERATION

Conversions can be specified to occur once on any selected channel (single mode), or in a continuous cycle on a single channel or on a pre-determined group of channels (scan mode). A maximum of 4 channels can be chosen for continuous cyclic scanning until the user software resets the ADST bit of the A/D Control Status Register (ADCSR) to 0, at which point the process is halted. This mode is useful if the application requires constant monitoring of external analog voltages. The conversion process can be initiated either in hardware or by the user software. The hardware approach starts the conversion by detecting a falling edge at the ADTRG- pin, while the software method requires the ADST bit of the ADCSR be set to 1. Upon conversion completion, the option of requesting an interrupt (ADI) is offered by setting the interrupt enable bit ADIE to 1; however, this interrupt has the lowest priority level in the exception map. Also, bit ADF is set at the end of the conversion; the user software must reset this bit shortly thereafter so it can indicate the end of the next conversion. The binary result is stored into one of the 4 data registers ADDR A-D, and is available to the user for further processing. Figures 1 and 2 show an example depicting the A/D converter operation under both single (channel 0) and scan mode (channels 0-2).

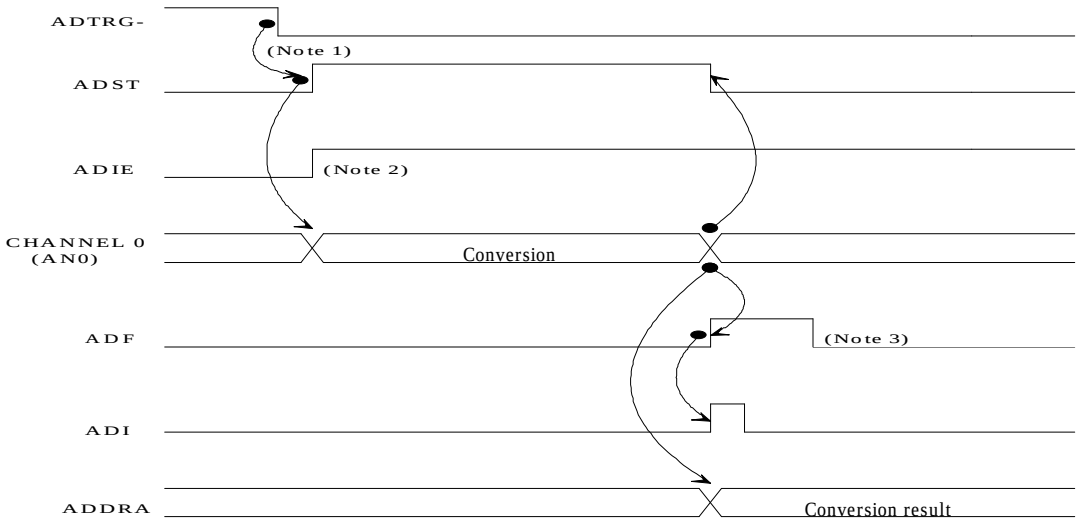

Figure 1. A/D Conversion in the single mode using channel 0.

Note 1: If the conversion process is started by the user software (ADST set to 1), disregard ADTRG-. If the hardware method is used, the falling edge at the ADTRG- pin will set the ADST bit to 1 automatically.
Notes 2 and 3: ADIE is set by the user software; ADF is set by the chip's hardware and reset by the user software.

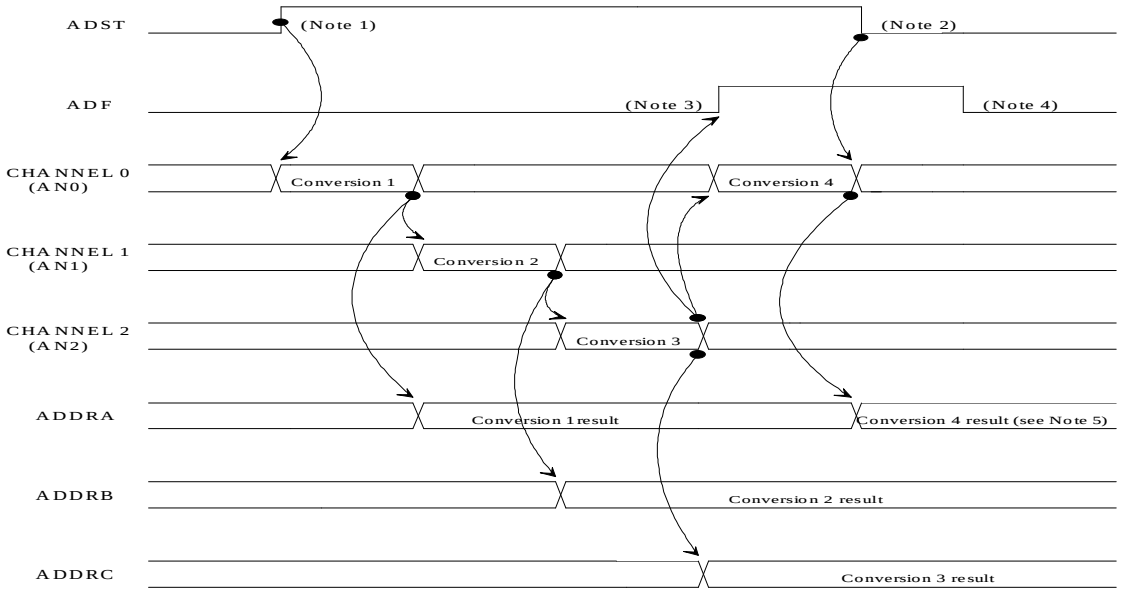

Figure 2. A/D Conversion in the scan mode using channels 0-2.

Notes 1 and 2: The duration of the conversion can be set for 2 fixed states, and is determined by the user software via bit CKS in the ADCSR. Bit ADST is high throughout this whole time.
Notes 3 and 4: After the third conversion of the first scan cycle is completed, bit ADF is automatically set high. The user software must clear this bit **before** the next scan cycle is completed, since it will be set high after each subsequent scan cycle. However, in this example, the conversion is terminated before completing the next full scan cycle, and the ADF bit is cleared shortly afterwards.
Note 5: The contents of ADDRA are automatically replaced following the end of conversion 4; therefore, the result of conversion 1 has to be processed between the end of conversion 1 and the end of conversion 4 in order for the data not to be lost.

The time of conversion can be set either for 16 system clock cycles per bit or to 8 system clock cycles per bit. Since a system clock cycle is composed of 2 Φ-states, the user has the choice to allow for a conversion time of maximum 242 states or 122 states respectively. This option is exercised by programming bit CKS of ADCSR as desired.

APPLICATION EXAMPLE

As always, the best way of learning is by program example. The H8/330 evaluation board is used as a target system, and the chip will operate in mode 3 (single-chip mode) since there is no need for external memory. Please refer to the H8/330 Evaluation Board User's Manual for a complete description of its characteristics. An external board is providing analog voltages at the channel 0 pin (AN0) through a simple voltage divider circuit that is activated by tweaking a variable potentiometer; a range of 0.77V - 5V is possible. The A/D conversion is initiated externally as well, by pressing a manual switch that causes a falling edge at the ADTRG- pin (on the H8/330 evaluation board) to occur. Two 74HC14 Schmitt Trigger inverters are used to provide a clean high-to-low transition. Figure 3 shows this configuration. The hexadecimal value of the conversion result is displayed on an ASCII data terminal. Table 1 shows the actual voltages corresponding to all possible hex conversion results.

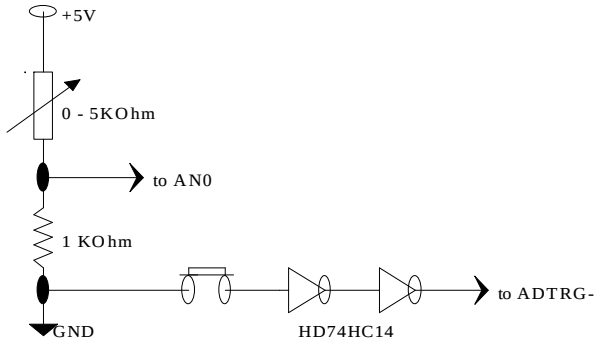

Figure 3. External board configuration.

PROGRAM DESCRIPTION

Following the usual system initializations, the SCI module is initialized for a transmission rate of 9600 baud and using a clock of 10MHz. The A/D module is set for single mode conversion, and channel 0 is chosen to sense the analog voltage. Bit ADIE of the ADCSR is set to 1 in order to enable the end-of-conversion interrupt service routine. Bit 0 of I/O Port 9, corresponding to the ADTRG- pin, is set to 1 and the program loops indefinitely waiting for the manual switch to be activated. Upon detecting a falling edge at the ADTRG- pin, the A/D module starts the actual conversion process. Bit ADF of the ADCSR will be automatically set to 1 upon completion and, since bit ADIE was set to 1 during initialization, the program counter will be loaded with the address of the first instruction of the interrupt service routine. The exception starts by clearing bit ADF, moving the conversion result into a general 8-bit register, and shifting the first digit into the lower nibble. In order to obtain the correct ASCII code for the digit, a hex value of 30 is added to the actual digit if the value falls between 0 and 9; if the digit lies between A and F, a value of 37 is added instead. Then, the ASCII equivalent of the first digit is transmitted asynchronously via the TxD pin to the data terminal that will display it. Afterwards, the hex result of the conversion is loaded into the general register again, and the exception proceeds to convert to ASCII and transmit the second digit of the conversion result. Figures 4 and 5 display the program flowcharts.

	0x	1x	2x	3x	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
x0	0.019	0.332	0.645	0.957	1.270	1.582	1.895	2.207	2.520	2.832	3.145	3.457	3.770	4.082	4.395	4.707
x1	0.039	0.352	0.664	0.977	1.289	1.602	1.914	2.227	2.539	2.852	3.164	3.477	3.789	4.102	4.414	4.727
x2	0.058	0.371	0.684	0.996	1.309	1.621	1.934	2.247	2.559	2.871	3.184	3.496	3.809	4.121	4.434	4.746
x3	0.078	0.391	0.703	1.016	1.328	1.641	1.953	2.267	2.578	2.891	3.203	3.516	3.828	4.141	4.453	4.766
x4	0.097	0.410	0.723	1.035	1.348	1.660	1.973	2.285	2.598	2.910	3.223	3.535	3.848	4.160	4.473	4.785
x5	0.117	0.430	0.742	1.055	1.367	1.680	1.992	2.305	2.617	2.930	3.242	3.555	3.867	4.180	4.492	4.805
x6	0.137	0.449	0.762	1.074	1.387	1.699	2.012	2.324	2.637	2.949	3.262	3.574	3.887	4.199	4.512	4.824
x7	0.156	0.469	0.781	1.094	1.406	1.719	2.031	2.344	2.656	2.969	3.281	3.594	3.906	4.219	4.531	4.844
x8	0.176	0.488	0.801	1.113	1.426	1.738	2.051	2.363	2.676	2.988	3.301	3.613	3.926	4.239	4.551	4.863
x9	0.195	0.508	0.820	1.133	1.445	1.758	2.070	2.383	2.695	3.008	3.320	3.633	3.945	4.258	4.570	4.883
xA	0.215	0.527	0.840	1.152	1.465	1.777	2.090	2.402	2.715	3.027	3.340	3.652	3.965	4.277	4.590	4.902
xB	0.234	0.547	0.859	1.172	1.484	1.797	2.109	2.422	2.734	3.047	3.359	3.672	3.984	4.297	4.609	4.922
xC	0.254	0.566	0.879	1.191	1.504	1.816	2.129	2.441	2.754	3.066	3.379	3.691	4.004	4.316	4.629	4.941
xD	0.273	0.586	0.898	1.211	1.523	1.836	2.148	2.461	2.773	3.086	3.398	3.711	4.023	4.336	4.648	4.961
xE	0.293	0.605	0.918	1.230	1.543	1.855	2.168	2.480	2.793	3.105	3.418	3.730	4.043	4.355	4.668	4.980
xF	0.313	0.625	0.938	1.250	1.562	1.875	2.188	2.500	2.813	3.125	3.438	3.750	4.063	4.375	4.688	5.000

TABLE 1. Conversion result hex value correspondence to analog voltage

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

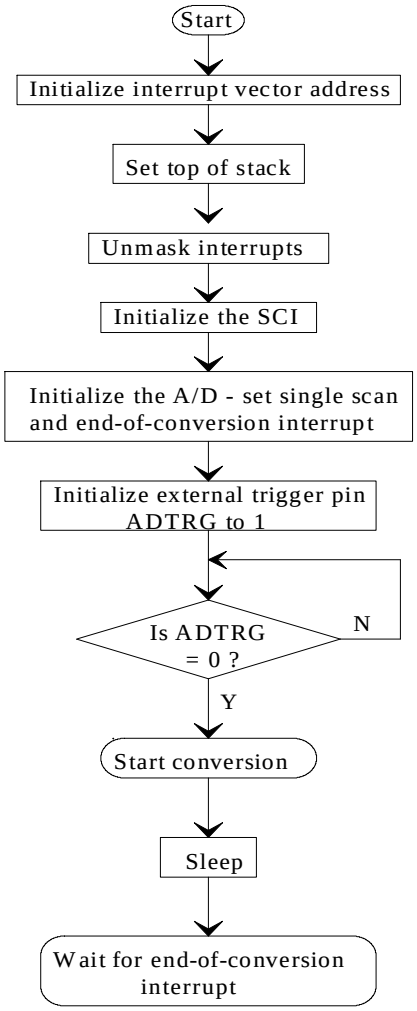

Figure 4. Main Program.

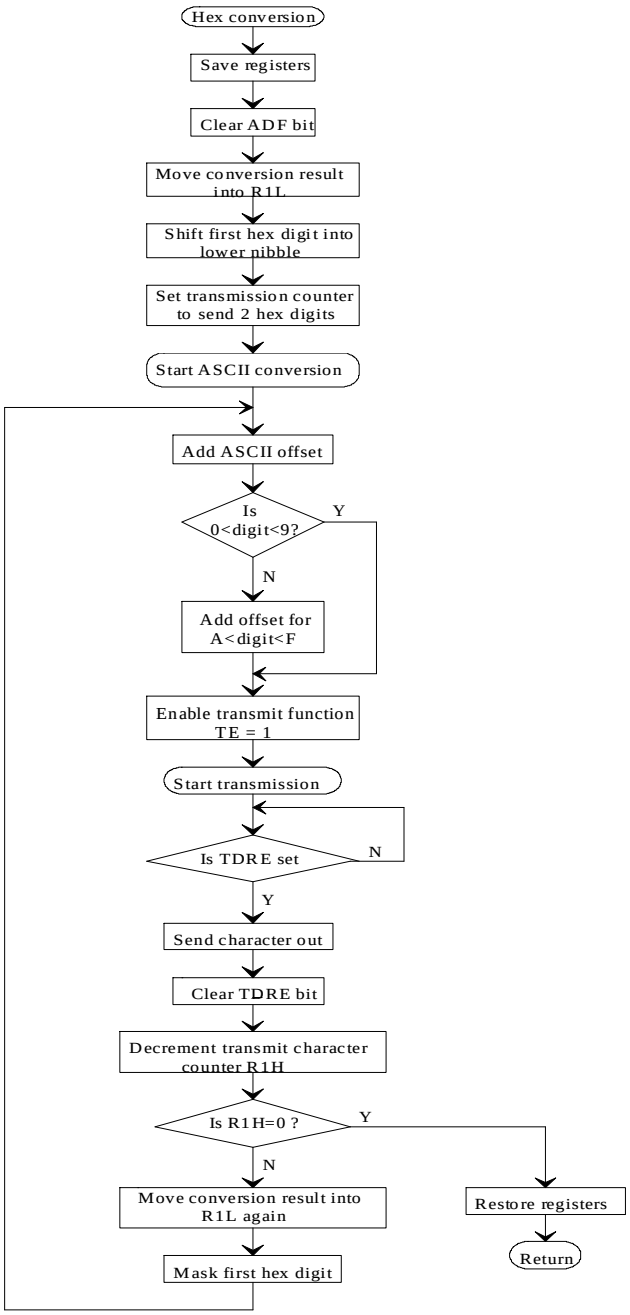

Figure 5. Interrupt Service Routine.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
Microtec Research ASMH83    Version 1.0A    Feb 03 10:38:44 1993     Page 1


Command line:  C:\MRI\ASMH83\ASMH83.EXE -l ad.src
Line        Addr
1                                   ;       Include the H8300 register definitions file
2
3                                      .include "C:\MRI\ASMH83\H8300.EQU"
4
5                                   ;       Specify interrupt vector address and program start
6
7                                      .org            H'3C
8           003C 052A                  .data.w         HEX_CONV
9                                      .org            H'500
10
11                                  ;       Initialize the stackpointer and mask all interrupts
12
13          0500 7907 FF80             mov.w           #TOP_RAM,R7             ;initialize stackpointer to H'FF80
14          0504 0700                  ldc             #0,CCR                  ;unmask all interrupts
15
16                                  ;       Initialize the SCI - set baud rate, communication format, and serial clock rate
17
18          0506 7FDA 7250             bclr            #5,@SCI0_SCR            ;disable the transmit function (clear TE bit)
19          050A F81F                  mov.b           #31,R0L                 ;set the BRR value corresponding to an internal
20          050C 38D9                  mov.b           R0L,@SCI0_BRR           ;clock of 10MHz and 9600 baud
21          050E F804                  mov.b           #H'04,R0L               ;set communication format and the clock rate
22          0510 38D8                  mov.b           R0L,@SCI0_SMR
23
24                                  ;       Initialize the A/D - specify the scanning mode, analog input channel,
25                                  ;                            and enable the external trigger signal
26
27          0512 F840                  mov.b           #H'40,R0L               ;set scan mode for channels AN0 and AN1 and
28          0514 38E8                  mov.b           R0L,@AD_ADCSR           ;enable end-of-conversion interrupt request
29          0516 7FEA 7070             bset            #7,@AD_ADCR             ;enable external trigger
30          051A 7FC1 7000             bset            #0,@P9_DR               ;initialize ADTRG- trigger input to 1
31
32                                  ;       Wait until A/D conversion is initiated by the manual reset switch
33
34                                  WAIT:
35          051E 7EC1 7300             btst            #0,@P9_DR               ;is the ADTRG- pin set to 0?
36          0522 46FA                  bne             WAIT                    ;if not, test again
37
38                                  ;       Perform the conversion
39
40                                  START_CONV:
41          0524 0180                  sleep                                   ;wait for end-of-conversion interrupt

42          0526 0000                  nop
43          0528 40FA                  bra             START_CONV
44
45                                  ;       Start HEX_CONV interrupt service routine - convert the 2 hex digits of the
46                                  ;       A/D result into ASCII code, and display the value onto the data terminal

47
48                                  HEX_CONV:
49          052A 6DF1                  push            R1                      ;save register
50          052C 7FE8 7270             bclr            #7,@AD_ADCSR            ;clear bit ADF
51          0530 29E0                  mov.b           @AD_ADDRA,R1L           ;put conversion hex result into R1L
52          0532 1109                  shlr            R1L                     ;shift first hex digit into lower nibble
53          0534 1109                  shlr            R1L
54          0536 1109                  shlr            R1L
55          0538 1109                  shlr            R1L
56          053A F102                  mov.b           #2,R1H                  ;set transmission counter to send the 2 hex
57                                                                                  ;digits
58                                  ASCII:
59          053C 8930                  add.b           #H'30,R1L               ;add ASCII offset
60          053E A93A                  cmp.b           #":",R1L                ;is number between 0 and 9?
61          0540 4502                  bcs             TRANSMIT                ;if yes, transmit the hex digit
62          0542 8907                  add.b           #7,R1L                  ;if number between A and F, add offset
63
64                                  ;       Transmit the conversion result hex digits
65
66                                  TRANSMIT:
67          0544 7FDA 7050             bset            #5,@SCI0_SCR            ;enable the transmit function (set TE bit)
68          0548 7EDC 7370             btst            #7,@SCI0_SSR            ;is the TDRE bit set?
69          054C 47F6                  beq             TRANSMIT                ;if not, loop until TDRE=1
70          054E 39DB                  mov.b           R1L,@SCI0_TDR           ;if yes, send the character data out
71          0550 7FDC 7270             bclr            #7,@SCI0_SSR            ;clear the TDRE bit
72          0554 1A01                  dec             R1H                     ;decrement transmit counter
73          0556 4706                  beq             END_TRANS               ;if transmit counter is 0, go to END_TRANS
74                                                                                  ;if not, proceed to convert to ASCII and
transmit
75                                                                                  ;the last hex digit
76          0558 29E0                  mov.b           @AD_ADDRA,R1L           ;put conversion hex result into R1L again
77          055A E90F                  and.b           #H'0F,R1L               ;mask first hex digit
78          055C 40DE                  bra             ASCII                   ;go to ASCII and repeat the process for the last
79                                                                                  ;hex digit
80                                  END_TRANS:
81          055E 6D71                  pop             R1                      ;restore register
82          0560 5670                  rte
83                                     .end
```