Hitachi America, Ltd.

Application Note

H8/338

Cristian Tomescu

D/A Conversion Using H8/338

INTRODUCTION

The H8/338, along with the other members of its family (the H8/337 and H8/336) have a built-in dual channel digital-to-analog module. Its capability of generating software-programmed analog voltage levels, with an 8-bit resolution between 0 and AVcc, makes this device useful in a variety of applications (such as servo DC-motor control). This document will describe in detail the D/A unit operation, and will exemplify its usefulness through one of the many possible "real-life" applications. A complete description of the hardware setup as well as code flowcharts and the program listing completes this application note.

DESCRIPTION OF OPERATION

The D/A module has 2 available channels that can be either operated independently or controlled together. The D/A Output Enable, bits 7 and 6 of the D/A Control Register (DACR), enable (or disable) conversion on channel 1 and 0 respectively. When set, the D/A Enable (bit 5 of the same register) causes simultaneous D/A conversion at both channels; **even if one of the 2 output enable bits is cleared to 0.** However, if both bits 7 and 6 are cleared, setting bit 5 will not make any difference. The 2 D/A Data Registers, DADR0 and DADR1, are used to store the binary value which is to be converted into an analog voltage via channel 0, and channel 1 outputs respectively. These registers can hold 8 bits of information, which means that, for a given analog output range, they can convert up to a maximum of 256 different voltage levels. Since the H8/338 microcontroller sub-family requires an analog supply voltage (AVcc) of 5V with a maximum 10% variation from the nominal value, the analog output resolution will be 5/256 +/- 10% = 19.5mV +/- 10%. Therefore, the error margin for any desired analog output voltage will not be higher than 19.5/2 = 9.75mV +/- 10%. An important fact to be noted is that **AVcc must be 5V even if the device operates from a 3V Vcc power supply**. The actual conversion process is relatively simple. First the user must load the appropriate data register(s) with the hexadecimal data corresponding to the desired output voltage. Second, the user must set the appropriate control bit(s) as described in the first paragraph above to start the conversion process. The time it takes for the D/A conversion is specified to be less than or equal to 10us. At the end of the conversion, I/0 Port 7 bits 6 and 7, normally used as input lines, automatically switch to analog outputs DA0 and DA1, depending upon which channel was enabled. If the output enable bit(s) are not cleared, the voltage level at the activated channel(s) is kept until a different hex value is written into the correponding data register. The time lag between when a new value is loaded into a data register and the moment conversion starts is less than 3 T-states of the internal clock. When the appropriate output enable bit(s) is cleared to 0, the correponding D/A output becomes an input pin again. This process is illustrated at channel 0 in figure 1 below.

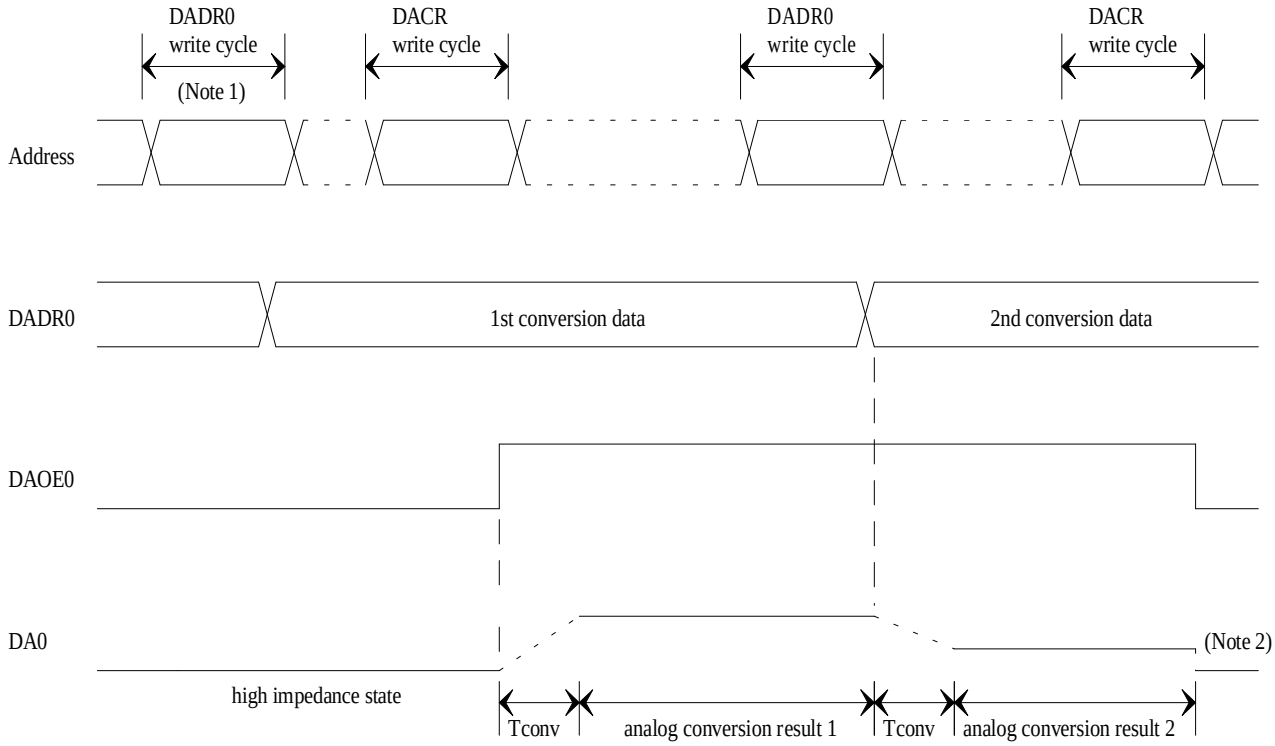

Figure 1. D/A Conversion on channel 0.

Note 1: Writing the D/A registers is done in a 3 T-state internal clock cycle.

Note 2: When the D/A output enable bit (DAOE0) is cleared back to 0, the analog output (DA0) becomes a digital input pin, and its value will depend upon the previous analog level at the pin. Specifically, if the analog voltage fell into the range between 3.5 - 5V, the input level will be high, but if the voltage range was between 0 - 0.75V, the input level will be low. Outside the above mentioned ranges, the input level cannot be predicted.

APPLICATION EXAMPLE

As mentioned in the introduction, one of the most common uses for a built-in D/A converter is for controlling the operation of low power DC motors. Specifically, there is a direct relationship between the speed of the motor and the analog voltage supplied to its coils. For instance, since the H8/338 sub-family can output up to 256 different voltage levels within the 0-5V range, the rotation speed of a fan can be minutely controlled. The presence of 2 separate analog channels expands the control capability of the chip to 2 servo controlled devices.

As an example, we could use the H8/338 to simulate the rotational velocity of 2 small fans. Each fan will be capable of rotating at 8 different speeds, which will be determined by a command entered by an operator on a control keyboard. The speeds of rotation will be slightly different for each fan. The following table indicates the corresponding 8 analog voltage levels at each control channel:

DA0 (V)	DA1 (V)
0.3	0.6
0.9	1.3
1.6	1.9
2.2	2.5
2.8	3.1
3.4	3.8
4.1	4.4
4.7	5.0

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

Such a control unit would require a small circuit board comprised of the H8/338 set in any mode of operation, an external EPROM to hold the code (if the chip is set for mode 2), and a RS-232 interface to provide the asynchronous communication channel between the operator console and the control system. The system will consist of a small keyboard and a display terminal. A self explanatory menu will be shown on the screen, and the operator will enter the appropriate command to adjust the fan speed as desired. However, if the data entered by the operator does not correspond to any of the analog voltages shown on the display terminal, the display terminal screen will be refreshed with a warning menu allowing the operator to type in an allowed value. The 2 fans will be connected to the control board through the analog voltage output pins DA0 and DA1.

In this example, we used the E3000 emulator box to act as the H8/338 microcontroller. Since the H8/338 is pin-by-pin and functionally compatible with the H8/330, sharing the same memory and register space addresses, the H8/330 evaluation board was used as the target system. Please refer to the H8/330 Evaluation Board User's Manual for a complete description of its features.

The S-record control code was downloaded from the PC to the E3000 emulator memory, and the E3000 CPU mode was set to single-chip. An additional board was used to provide the RS-232 communication channel between the transmit and receive lines of the chip, and the display terminal input. Figure 3 shows the schematic of this SCI interface board. The control menu is transmitted to a VT320 terminal via the TxD output pin. Menu transmission is initiated by pressing the push-button switch SW1. As a result, a sharp falling edge will appear at the IRQ1- pin, since the signal is driven through 2 Schmitt-trigger inverters. The RxD input pin will be used to receive the voltage data entered by the control keyboard. The requested voltage value will be ehoed on the display terminal as well. A MAX232 Dual-Channel Driver/Receiver is employed to provide the necessary voltage levels for proper RS-232 conversation. After a short execution time, the chosen analog level will be present at one of the 2 D/A channels. In this example, we chose to control only one fan at a time in order to provide for different individual speeds. Figure 2 illustrates the setup used.

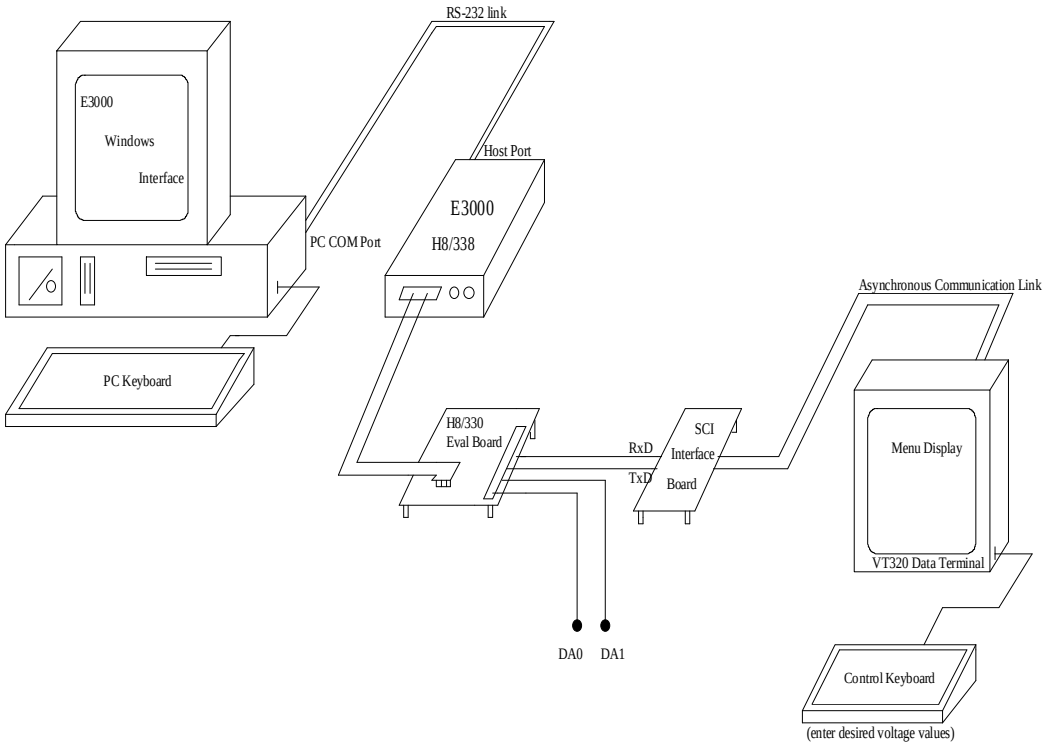

Figure 2. System Connection.

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

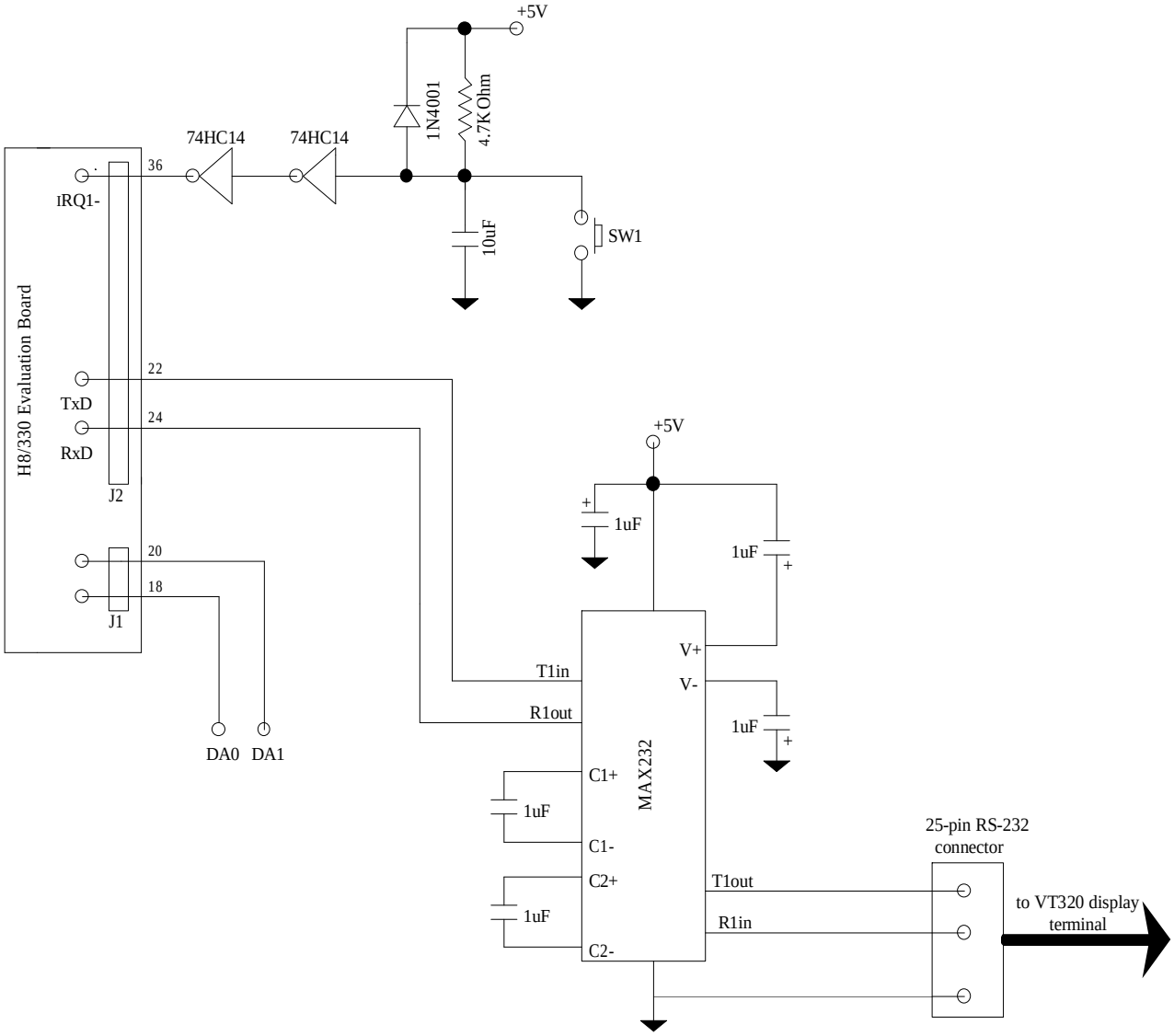

Figure 3. SCI Board schematic.

PROGRAM DESCRIPTION

The program listing starts on page 11. The interrupt vector table is initialized, followed by the menu's ASCII data strings. Then the MCU unit is initialized by setting the stackpointer and unmasking all interrupts. The SCI module is initialized for asynchronous transmission, and a baud rate of 9600 operating from a 10MHz internal clock source. The transmit and receive functions are enabled along with IRQ1. Afterwards, the CPU will enter sleep mode waiting for an interrupt to occur. The start menu data string transmission will be initiated by a falling edge at the eval board's IRQ1- pin. As a result, the first string character will be transmitted to the display terminal, and IRQ1 will be disabled. The rest of the characters will be transmitted one-by-one repeatedly using a service routine initiated by the transmission-end interrupt (TxI). The end of the transmission is detected by a null character at the end of the menu data string, at which time the TxI will be disabled and the receive-end interrupt (RxI) will be enabled. Figure 4 shows the start menu and the available voltage choices as they appear on the terminal screen.

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

TAKE YOUR PICK

FAN A (DA0)	FAN B (DA1)
0.3	0.6
0.9	1.3
1.6	1.9
2.2	2.5
2.8	3.1
3.4	3.8
4.1	4.4
4.7	5.0

ENTER VALUE FOLLOWED BY "!" SIGN:

Figure 4. Start Menu and available voltages.

The CPU will re-enter sleep mode until the user starts entering one of the above voltage data. If the entered data does not correpond to the menu values, bit 0 of I/O Port 8 will be used to provide a falling edge at the IRQ2- pin, hence activating a new service routine. This routine will transmit a second menu (warning menu) to the display screen, telling the user to choose again from the pre-determined choices. Figure 5 below shows this menu as it will appear on the screen:

THIS CHOICE IS NOT ALLOWED.
PLEASE ENTER ONE OF THE VALUES BELOW, FOLLOWED BY AN "!" SIGN:

FAN A (DA0)	FAN B (DA1)
0.3	0.6
0.9	1.3
1.6	1.9
2.2	2.5
2.8	3.1
3.4	3.8
4.1	4.4
4.7	5.0

VALUE:

Figure 5. Warning menu.

The ASCII data corresponding to the requested analog voltage is placed into a pre-defined memory block location, and the first digit value is determined through a series of compare operations. Next, the second digit value is found using the same method and, for each analog voltage value, the corresponding D/A hex code is placed into one of the 2 D/A Data Registers (DADR0 or DADR1). Appropriately setting either bit 7 (DAOE1) or bit 6 (DAOE0) of the D/A Control Register (DACR) starts the conversion process. Alternatively, a look-up table method could have been used to determine the voltage data entered. After a delay time not longer than 10us, the desired voltage (within a +/- 10% accuracy) will be present at either DA1 or DA0 output pins. The CPU will then re-enter the sleep mode. If a different analog output is needed at either channel, the process can be started again by initiating IRQ1. Figures 6, 7, 8, 9, and 10 illustrate the event sequence flowcharts.

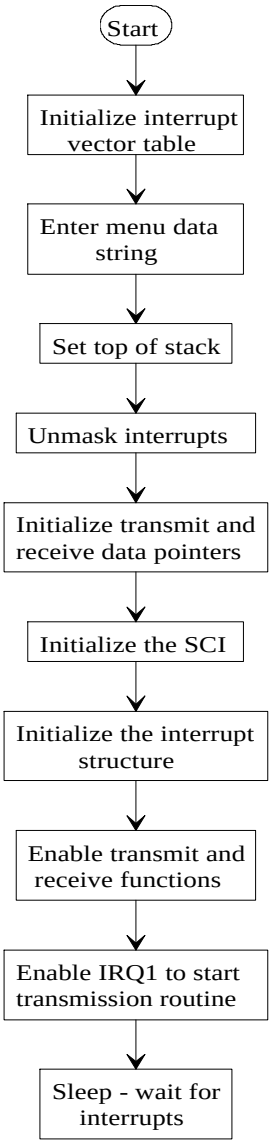

Figure 6. Main Program.

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

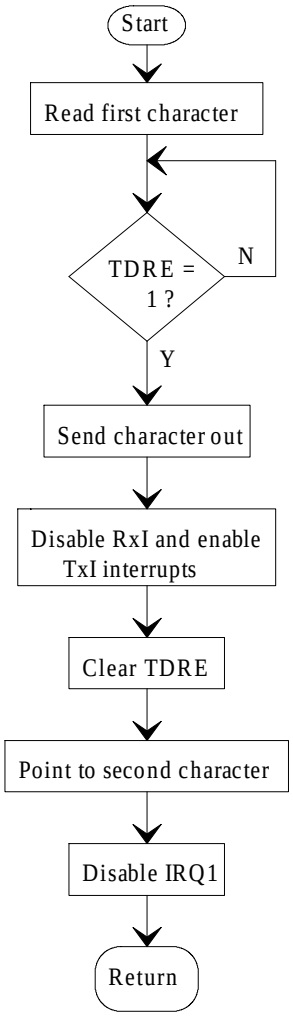

Figure 7. Start Menu Transmission Routine.

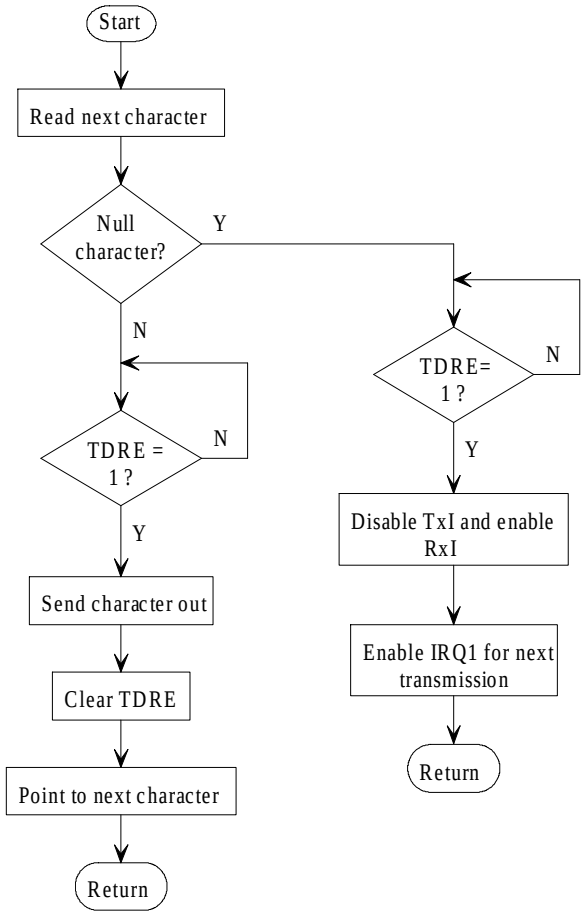

Figure 8. Transmit Interrupt Service Routine.

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

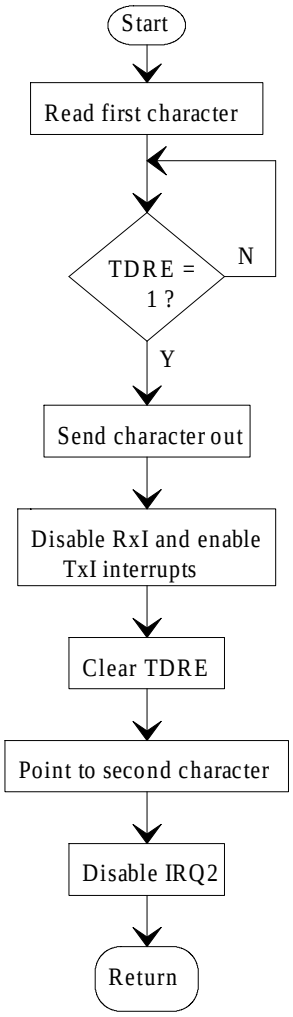

Figure 9. Warning Menu Transmission Routine.

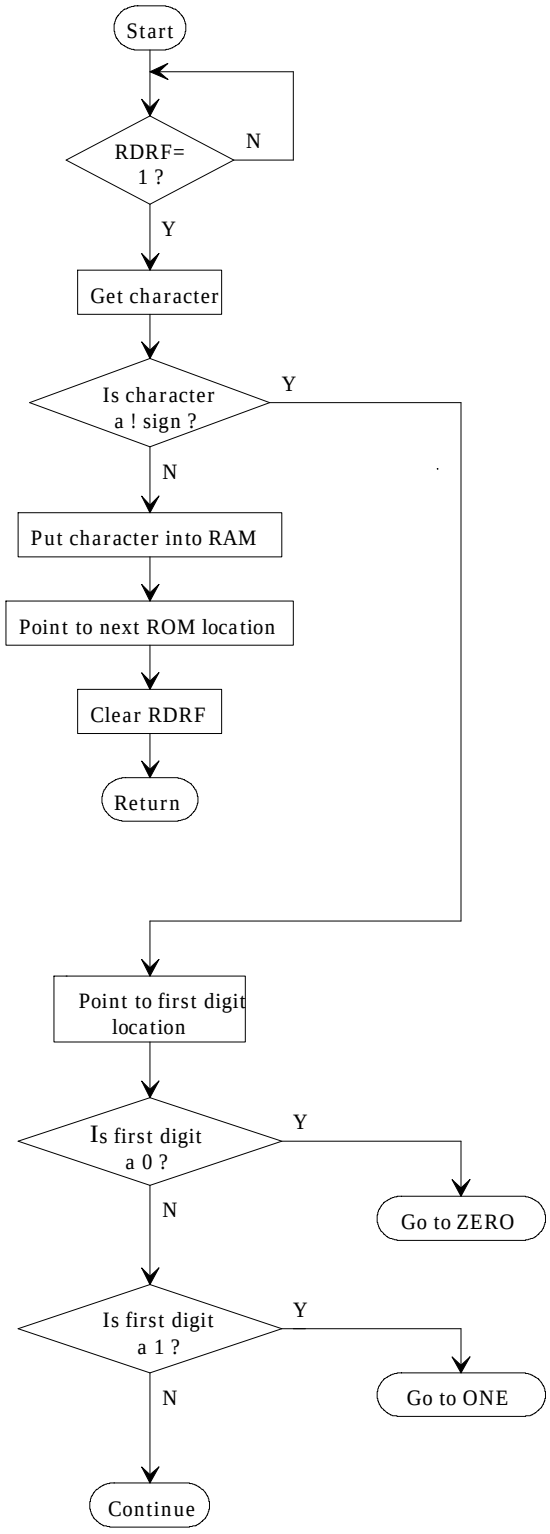

Figure 10. Receive Interrupt Service Routine and D/A Converter hex code generation.

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

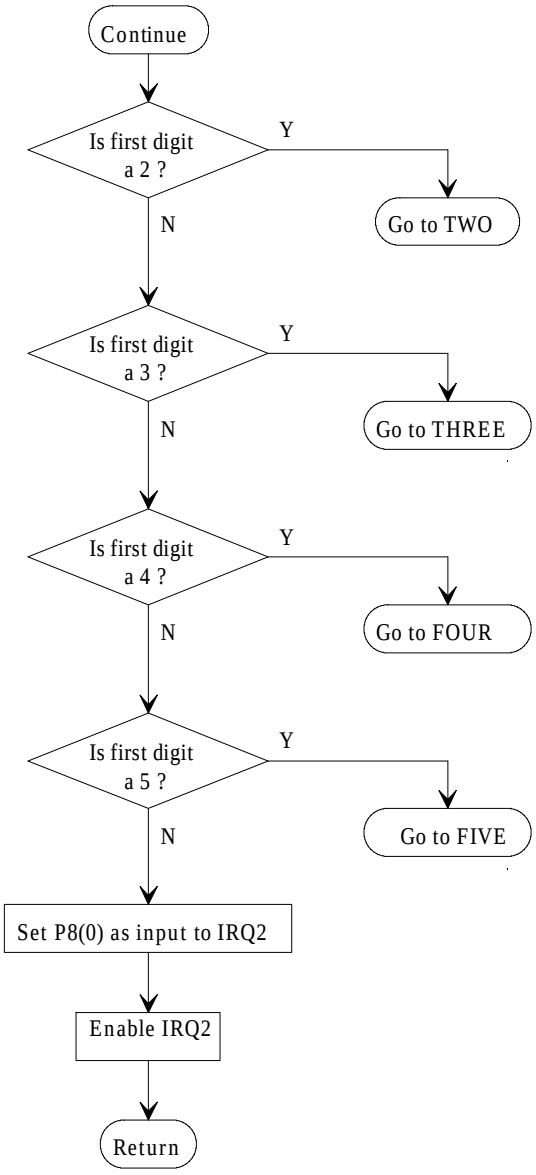

Figure 10. Continuation.

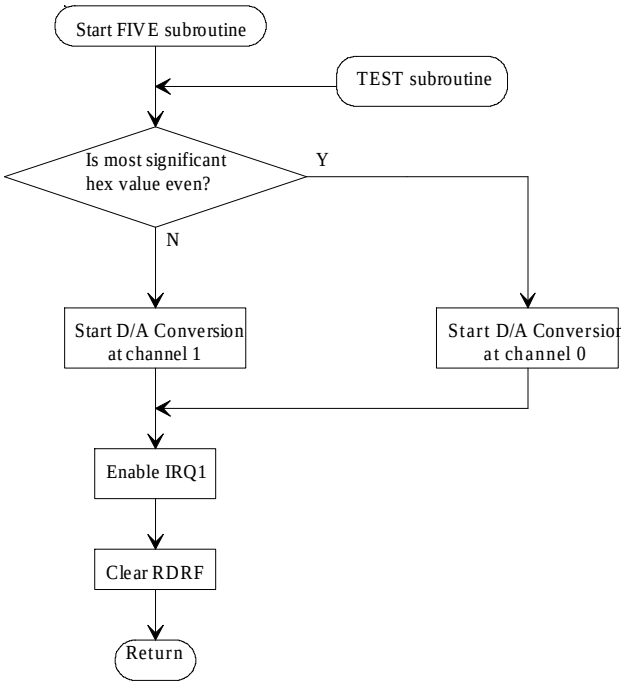

Figure 10. Continuation.

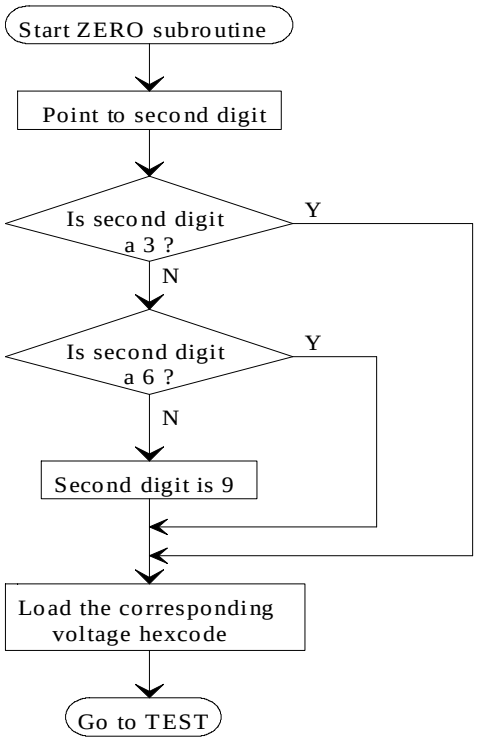

Figure 10. Continuation

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

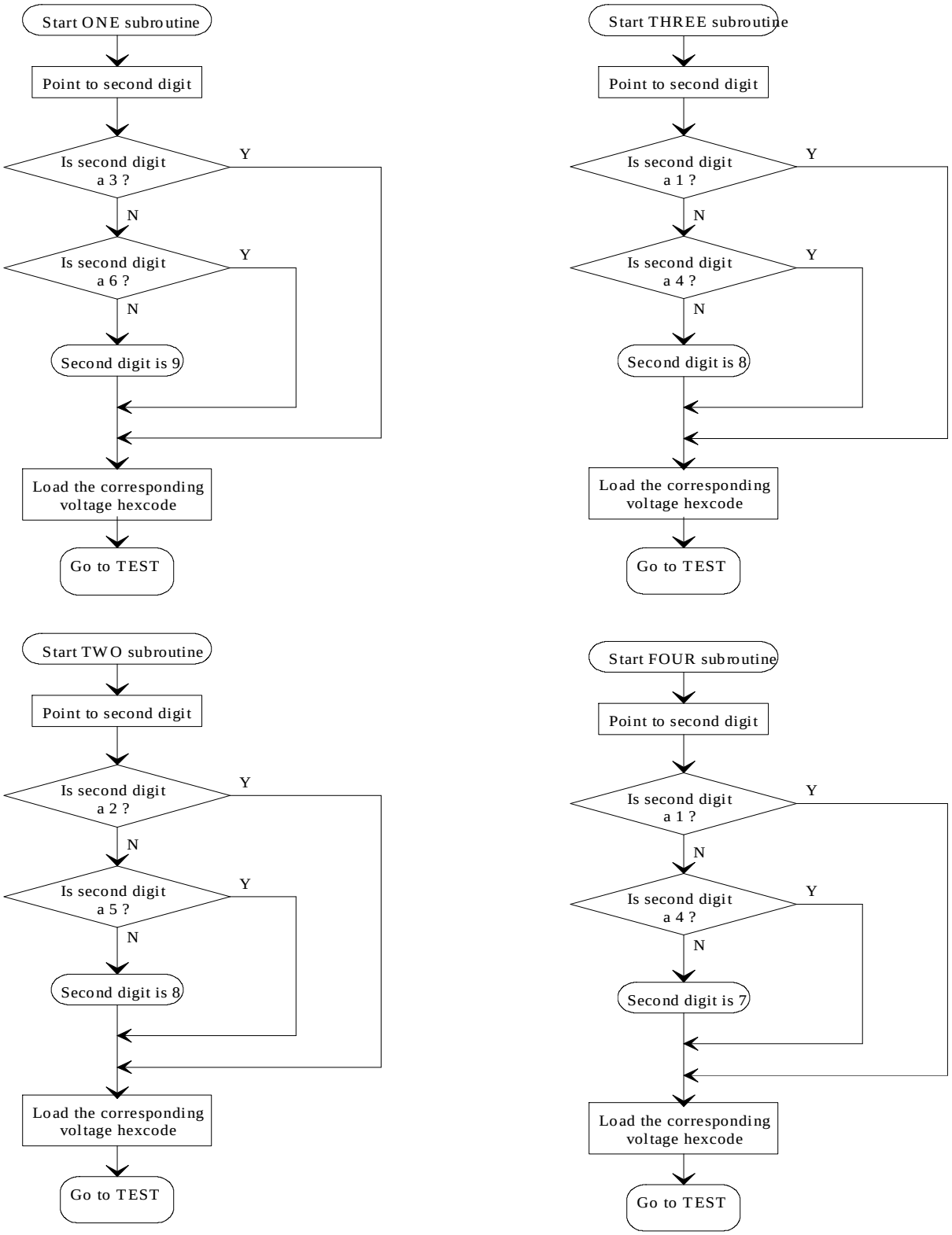

Figure 10. Continuation

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

LISTING OF THE H8/338.EQU INCLUDE FILE

```
.list off


;H8338 Register definitions
;Stack and data string addresses
TOP_RAM         .equ    H'FF80
MENU1_DATA      .equ    H'100
MENU2_DATA      .equ    H'900
DA_DATA         .equ    H'0FF0

;SCI register addresses
SCI0_SCR        .equ    H'FFDA
SCI0_SMR        .equ    H'FFD8
SCI0_BRR        .equ    H'FFD9
SCI0_SSR        .equ    H'FFDC
SCI0_TDR        .equ    H'FFDB
SCI0_RDR        .equ    H'FFDD

;Port addresses
P9_DDR          .equ    H'FFC0
P9_DR           .equ    H'FFC1
P8_DDR          .equ    H'FFBD
P8_DR           .equ    H'FFBF

;Interrupt controller register addresses
ISCR            .equ    H'FFC6
IER             .equ    H'FFC7

;D/A register addresses
DADR0           .equ    H'FFA8
DADR1           .equ    H'FFA9
DACR            .equ    H'FFAA

                .list on
```


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

```
Microtec Research ASMH83    Version 1.0A    Mar 12 11:22:35 1993     Page 1


Command line:  C:\MRI\ASMH83\ASMH83.EXE -l da.src
Line        Addr
1                                           ;       Include the H8/338 register definitions file
2
3                                           .include "c:\mri\asmh83\h8338.equ"
4
5                                           ;       Specify interrupt vector addresses
6
7                                           .org            H'0A
8           000A 103E                       .data.w         START_MENU1             ;use IRQ1 to initiate start menu tr
                                    ansmission
9                                           .org            H'0C
10          000C 1060                       .data.w         START_MENU2             ;use IRQ2 to initiate warning menu
                                    transmission
11                                          .org            H'38
12          0038 10AE                       .data.w         REC_HEX                 ;use RxI to receive desired voltage
                                     value
13                                          .org            H'3A
14          003A 1086                       .data.w         MENU_CONT               ;use TxI to transmit menu
15
16                                          ;       Specify location of start menu ASCII character string
17
18                                          .org            H'100
19          0100 =01B0=of=                  .datab.b        432,H'20
                 20
20          02B0 5441 4B45 2059 4F55        .data.b         H'54,H'41,H'4B,H'45,H'20,H'59,H'4F,H'55
21          02B8 5220 5049 434B             .data.b         H'52,H'20,H'50,H'49,H'43,H'4B
22          02BE =0086=of=                  .datab.b        134,H'20
                 20
23          0344 4641 4E20 41               .data.b         H'46,H'41,H'4E,H'20,H'41
24          0349 2028 4441 3029             .data.b         H'20,H'28,H'44,H'41,H'30,H'29
25          034F =001E=of=                  .datab.b        30,H'20
                 20
26          036D 4641 4E20 42               .data.b         H'46,H'41,H'4E,H'20,H'42
27          0372 2028 4441 3129             .data.b         H'20,H'28,H'44,H'41,H'31,H'29
28          0378 =001F=of=                  .datab.b        31,H'20
                 20
29          0397 302E 33                    .data.b         H'30,H'2E,H'33
30          039A =0026=of=                  .datab.b        38,H'20
                 20
31          03C0 302E 36                    .data.b         H'30,H'2E,H'36
32          03C3 =0024=of=                  .datab.b        36,H'20
                 20
33          03E7 302E 39                    .data.b         H'30,H'2E,H'39
34          03EA =0027=of=                  .datab.b        39,H'20
                 20
35          0411 312E 33                    .data.b         H'31,H'2E,H'33
36          0414 =0024=of=                  .datab.b        36,H'20
                 20
37          0438 312E 36                    .data.b         H'31,H'2E,H'36
38          043B =0027=of=                  .datab.b        39,H'20
                 20
39          0462 312E 39                    .data.b         H'31,H'2E,H'39
40          0465 =0024=of=                  .datab.b        36,H'20
                 20
41          0489 322E 32                    .data.b         H'32,H'2E,H'32
42          048C =0027=of=                  .datab.b        39,H'20
                 20
43          04B3 322E 35                    .data.b         H'32,H'2E,H'35
44          04B6 =0024=of=                  .datab.b        36,H'20
                 20
45          04DA 322E 38                    .data.b         H'32,H'2E,H'38
46          04DD =0027=of=                  .datab.b        39,H'20
                 20
47          0504 332E 31                    .data.b         H'33,H'2E,H'31
48          0507 =0024=of=                  .datab.b        36,H'20
                 20
49          052B 332E 34                    .data.b         H'33,H'2E,H'34
50          052E =0027=of=                  .datab.b        39,H'20
                 20
51          0555 332E 38                    .data.b         H'33,H'2E,H'38
52          0558 =0024=of=                  .datab.b        36,H'20
                 20
53          057C 342E 31                    .data.b         H'34,H'2E,H'31
54          057F =0027=of=                  .datab.b        39,H'20
                 20
55          05A6 342E 34                    .data.b         H'34,H'2E,H'34
56          05A9 =0024=of=                  .datab.b        36,H'20
                 20
57          05CD 342E 37                    .data.b         H'34,H'2E,H'37
58          05D0 =0027=of=                  .datab.b        39,H'20
                 20
59          05F7 352E 30                    .data.b         H'35,H'2E,H'30
60          05FA =005E=of=                  .datab.b        94,H'20
                 20
61          0658 454E 5445 5220             .data.b         H'45,H'4E,H'54,H'45,H'52,H'20
62          065E 5641 4C55 4520             .data.b         H'56,H'41,H'4C,H'55,H'45,H'20
63          0664 464F 4C4C 4F57 4544        .data.b         H'46,H'4F,H'4C,H'4C,H'4F,H'57,H'45,H'44
64          066C 2042 5920 2221 2220        .data.b         H'20,H'42,H'59,H'20,H'22,H'21,H'22,H'20
65          0674 5349 474E 3A00             .data.b         H'53,H'49,H'47,H'4E,H'3A,H'00
66
67                                          ;       Specify location of warning menu ASCII character string
68
69                                          .org            H'900
70          0900 =024B=of=                  .datab.b        587,H'20
                 20
```

```
Microtec Research ASMH83    Version 1.0A    Mar 12 11:22:35 1993     Page 2


Line        Addr
71          0B4B 5448 4953 2043 484F                    .data.b         H'54,H'48,H'49,H'53,H'20,H'43,H'48,H'4F,H'49
                 49
72          0B54 4345 2049 5320 4E4F                    .data.b         H'43,H'45,H'20,H'49,H'53,H'20,H'4E,H'4F,H'54
                 54
73          0B5D 2041 4C4C 4F57 4544                    .data.b         H'20,H'41,H'4C,H'4C,H'4F,H'57,H'45,H'44,H'2E
                 2E
74          0B66 2050 4C45 4153 4520                    .data.b         H'20,H'50,H'4C,H'45,H'41,H'53,H'45,H'20,H'45
                 45
75          0B6F 4E54 4552 204F 4E45                    .data.b         H'4E,H'54,H'45,H'52,H'20,H'4F,H'4E,H'45,H'20
                 20
76          0B78 4F46 2054 4845 2056                    .data.b         H'4F,H'46,H'20,H'54,H'48,H'45,H'20,H'56,H'41
                 41
77          0B81 4C55 4553 2042 454C                    .data.b         H'4C,H'55,H'45,H'53,H'20,H'42,H'45,H'4C,H'4F
                 4F
78          0B8A 5720 464F 4C4C 4F57                    .data.b         H'57,H'20,H'46,H'4F,H'4C,H'4C,H'4F,H'57,H'45
                 45
79          0B93 4420 4259 2041 4E20                    .data.b         H'44,H'20,H'42,H'59,H'20,H'41,H'4E,H'20,H'22
                 22
80          0B9C 2122 2053 4947 4E3A                    .data.b         H'21,H'22,H'20,H'53,H'49,H'47,H'4E,H'3A,H'00
                 00
81          0BA5 =00AB=of=                              .datab.b        171,H'20
                 20
82          0C50 4641 4E20 41                           .data.b         H'46,H'41,H'4E,H'20,H'41
83          0C55 2028 4441 3029                         .data.b         H'20,H'28,H'44,H'41,H'30,H'29
84          0C5B =001E=of=                              .datab.b        30,H'20
                 20
85          0C79 4641 4E20 42                           .data.b         H'46,H'41,H'4E,H'20,H'42
86          0C7E 2028 4441 3129                         .data.b         H'20,H'28,H'44,H'41,H'31,H'29
87          0C84 =001F=of=                              .datab.b        31,H'20
                 20
88          0CA3 302E 33                                .data.b         H'30,H'2E,H'33
89          0CA6 =0027=of=                              .datab.b        39,H'20
                 20
90          0CCD 302E 36                                .data.b         H'30,H'2E,H'36
91          0CD0 =0025=of=                              .datab.b        37,H'20
                 20
92          0CF5 302E 39                                .data.b         H'30,H'2E,H'39
93          0CF8 =0027=of=                              .datab.b        39,H'20
                 20
94          0D1F 312E 33                                .data.b         H'31,H'2E,H'33
95          0D22 =0025=of=                              .datab.b        37,H'20
                 20
96          0D47 312E 36                                .data.b         H'31,H'2E,H'36
97          0D4A =0027=of=                              .datab.b        39,H'20
                 20
98          0D71 312E 39                                .data.b         H'31,H'2E,H'39
99          0D74 =0025=of=                              .datab.b        37,H'20
                 20
100         0D99 322E 32                                .data.b         H'32,H'2E,H'32
101         0D9C =0027=of=                              .datab.b        39,H'20
                 20
102         0DC3 322E 35                                .data.b         H'32,H'2E,H'35
103         0DC6 =0025=of=                              .datab.b        37,H'20
                 20
104         0DEB 322E 38                                .data.b         H'32,H'2E,H'38
105         0DEE =0027=of=                              .datab.b        39,H'20
                 20
106         0E15 332E 31                                .data.b         H'33,H'2E,H'31
107         0E18 =0025=of=                              .datab.b        37,H'20
                 20
108         0E3D 332E 34                                .data.b         H'33,H'2E,H'34
109         0E40 =0027=of=                              .datab.b        39,H'20
                 20
110         0E67 332E 38                                .data.b         H'33,H'2E,H'38
111         0E6A =0025=of=                              .datab.b        37,H'20
                 20
112         0E8F 342E 31                                .data.b         H'34,H'2E,H'31
113         0E92 =0027=of=                              .datab.b        39,H'20
                 20
114         0EB9 342E 34                                .data.b         H'34,H'2E,H'34
115         0EBC =0025=of=                              .datab.b        37,H'20
                 20
116         0EE1 342E 37                                .data.b         H'34,H'2E,H'37
117         0EE4 =0027=of=                              .datab.b        39,H'20
                 20
118         0F0B 352E 30                                .data.b         H'35,H'2E,H'30
119         0F0E =005E=of=                              .datab.b        94,H'20
                 20
120         0F6C 454E 5445 5220                         .data.b         H'45,H'4E,H'54,H'45,H'52,H'20
121         0F72 5641 4C55 453A                         .data.b         H'56,H'41,H'4C,H'55,H'45,H'3A
122
123
124                                             ;       Specify program start address
125
126                                                     .org            H'1000
127
128                                             ;       Perform MCU initializations and save registers
129
130         1000 7907 FF80                              mov.w           #TOP_RAM,R7             ;set stackpointer to H'FF80
131         1004 0700                                   ldc             #0,CCR                  ;unmask all interrupts
132         1006 7906 0100                              mov.w           #MENU1_DATA,R6          ;point to the beginning address in
                                                                                                ;ROM of the start menu ASCII data string
133         100A 7905 0FF0                              mov.w           #DA_DATA,R5             ;point to ROM location of desired voltage
                                                                                                ;ASCII value
134
```


HITACHI

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

```
Microtec Research ASMH83    Version 1.0A    Mar 12 11:22:35 1993     Page 3


Line        Addr
135                                     ;       Initialize the SCI
136
137         100E 7FDA 7240                      bclr            #4,@SCI0_SCR            ;disable receive function (clear RE bit)
138         1012 7FDA 7250                      bclr            #5,@SCI0_SCR            ;disable transmit function (clear TE bit)
139         1016 F81F                           mov.b           #31,R0L                 ;set the BRR value corresponding to internal
140         1018 38D9                           mov.b           R0L,@SCI0_BRR           ;clock of 10MHz and 9600 baud
141         101A F804                           mov.b           #H'04,R0L               ;set communication format and clock rate
142         101C 38D8                           mov.b           R0L,@SCI0_SMR
143
144                                     ;       Initialize IRQ1 - it will be used to start the menu transmission routine
145
146         101E 7FC0 7210                      bclr            #1,@P9_DDR              ;P9(2) used as IRQ1 input
147         1022 7FC1 7010                      bset            #1,@P9_DR
148         1026 7FC6 7010                      bset            #1,@ISCR                ;IRQ1 will be triggered on falling edge
149
150                                     ;       Enable transmit and receive
151
152         102A 7FDA 7050                      bset            #5,@SCI0_SCR            ;enable transmission (set TE bit)
153         102E 7FDA 7040                      bset            #4,@SCI0_SCR            ;enable receiving (set RE bit)
154         1032 7FC7 7010                      bset            #1,@IER                 ;enable IRQ1
155
156                                     ;       Wait for start menu transmission interrupt to occur
157
158         1036 0180                   WAIT:   sleep
159         1038 0000                           nop
160         103A 0000                           nop
161         103C 40F8                           bra             WAIT
162
163                                     ;       Initiate start menu transmission routine
164
165                                     START_MENU1:
166         103E 6868                           mov.b           @R6,R0L                 ;get first character ASCII code
167         1040 7EDC 7370              TEST1:  btst            #7,@SCI0_SSR            ;is the TDRE bit set?
168         1044 47FA                           beq             TEST1                   ;if not, test again
169         1046 38DB                           mov.b           R0L,@SCI0_TDR           ;send the first character data out
170         1048 7FDA 7260                      bclr            #6,@SCI0_SCR            ;disable receive-end interrupt
171         104C 7FDA 7070                      bset            #7,@SCI0_SCR            ;enable transmit-end interrupt
172         1050 7FDC 7270                      bclr            #7,@SCI0_SSR            ;clear the TDRE bit
173         1054 0B06                           adds            #1,R6                   ;point to second character
174         1056 7FC7 7210                      bclr            #1,@IER                 ;disable IRQ1
175         105A 5670                           rte
176         105C 0000                           nop
177         105E 0000                           nop
178
179                                     ;       Initiate warning menu transmission routine
180
181                                     START_MENU2:
182         1060 7906 0900                      mov.w           #MENU2_DATA,R6          ;point to the beginning address of
                                                                                        ;the warning menu ASCII data string
183         1064 6868                           mov.b           @R6,R0L                 ;get first character ASCII code
184         1066 7EDC 7370              TEST2:  btst            #7,@SCI0_SSR            ;is the TDRE bit set?
185         106A 47FA                           beq             TEST2                   ;if not, test again
186         106C 38DB                           mov.b           R0L,@SCI0_TDR           ;send the first character data out
187         106E 7FDA 7260                      bclr            #6,@SCI0_SCR            ;disable receive-end interrupt
188         1072 7FDA 7070                      bset            #7,@SCI0_SCR            ;enable transmit-end interrupt
189         1076 7FDC 7270                      bclr            #7,@SCI0_SSR            ;clear the TDRE bit
190         107A 0B06                           adds            #1,R6                   ;point to second character
191         107C 7FC7 7220                      bclr            #2,@IER                 ;disable IRQ2
192         1080 5670                           rte
193         1082 0000                           nop
194         1084 0000                           nop
195
196
197
198                                     ;       Continue transmitting the menu
199
200         1086 6868                   MENU_CONT:
                                  mov.b           @R6,R0L                 ;get next character ASCII code
201         1088 4710                           beq             END_TRANS               ;if null character, go to END_TRANS
202         108A 7EDC 7370              TEST3:  btst            #7,@SCI0_SSR            ;is the TDRE bit set?
203         108E 47FA                           beq             TEST3                   ;if not, test again
204         1090 38DB                           mov.b           R0L,@SCI0_TDR           ;send the next character out
205         1092 7FDC 7270                      bclr            #7,@SCI0_SSR            ;clear the TDRE bit
206         1096 0B06                           adds            #1,R6                   ;point to next character
207         1098 5670                           rte
208                                     END_TRANS:
209         109A 7EDC 7370                      btst            #7,@SCI0_SSR            ;is last character transmitted?
210         109E 47FA                           beq             END_TRANS               ;if not, wait until finished
211         10A0 7FDA 7270                      bclr            #7,@SCI0_SCR            ;disable transmit-end interrupts
212         10A4 7FDA 7060                      bset            #6,@SCI0_SCR            ;enable receive-end interrupt
213         10A8 5670                           rte
214         10AA 0000                           nop
215         10AC 0000                           nop
216
217                                     ;       Start receiving the required voltage
218
219                                     REC_HEX:
220         10AE 7EDC 7360                      btst            #6,@SCI0_SSR            ;is the RDRF bit set?
```


HITACHI

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

```
Microtec Research ASMH83    Version 1.0A    Mar 12 11:22:35 1993     Page 4


Line        Addr
221         10B2 47FA                           beq             REC_HEX                 ;if not, test again
222         10B4 20DD                           mov.b           @SCI0_RDR,R0H           ;get hex value
223         10B6 A021                           cmp.b           #H'21,R0H               ;is character an exclamation sign?
224         10B8 470A                           beq             NEXT                    ;if so, end receiving
225         10BA 68D0                           mov.b           R0H,@R5                 ;put hex character into ROM location
226         10BC 0B05                           adds            #1,R5                   ;point to next ROM loacation
227         10BE 7FDC 7260                      bclr            #6,@SCI0_SSR            ;clear RDRF
228         10C2 5670                           rte
229
230                                     ;       Convert voltage ASCII code into corresponding D/A hex data, and perform the
                                        ;       conversion
231
232         10C4 7FDA 7260              NEXT:   bclr            #6,@SCI0_SCR            ;disable receive-end interrupt
233         10C8 1B05                           subs            #1,R5                   ;point to location of first digit
234         10CA 1B85                           subs            #2,R5
235         10CC 6850                           mov.b           @R5,R0H
236         10CE A01E                           cmp.b           #H'1E,R0H               ;is first digit a 0?
237         10D0 476A                           beq             BRANCH_0                ;if yes, go to ZERO
238         10D2 A01F                           cmp.b           #H'1F,R0H               ;is first digit a 1?
239         10D4 4768                           beq             BR_ONE                  ;if yes, go to ONE
240         10D6 A020                           cmp.b           #H'20,R0H               ;is first digit a 2?
241         10D8 4766                           beq             BR_TWO                  ;if yes, go to TWO
242         10DA A021                           cmp.b           #H'21,R0H               ;is first digit a 3?
243         10DC 4764                           beq             THREE                   ;if yes, go to THREE
244         10DE A022                           cmp.b           #H'22,R0H               ;is first digit a 4?
245         10E0 473A                           beq             FOUR                    ;if yes, go to FOUR
246         10E2 A023                           cmp.b           #H'23,R0H               ;is first digit a 5?
247         10E4 4718                           beq             FIVE                    ;if yes, go to FIVE
248         10E6 7FBD 7000              NONE:   bset            #0,@P8_DDR              ;since first or second digit is not allowed,
249         10EA 7FBF 7000                      bset            #0,@P8_DR               ;enable P8(0) as an input source for IRQ2
250         10EE 7FC6 7020                      bset            #2,@ISCR                ;IRQ2 will be activated on the falling edge
251         10F2 7FC7 7020                      bset            #2,@IER                 ;enable IRQ2
252         10F6 7FBF 7200                      bclr            #0,@P8_DR               ;falling edge at IRQ2
253         10FA 5670                           rte
254         10FC 0000                           nop
255         10FE F8FF                   FIVE:   mov.b           #H'FF,R0L               ;first digit is a 5
256         1100 7348                   TEST:   btst            #4,R0L                  ;is the most significant hex value even?
257         1102 4710                           beq             EVEN                    ;if yes, go to EVEN
258         1104 38A9                   ODD:    mov.b           R0L,@DADR1              ;put hex value into channel 1 data register
259         1106 7FAA 7070                      bset            #7,@DACR                ;start D/A conversion at channel 1
260                                     END_CONV:
261         110A 7FC7 7010                      bset            #1,@IER                 ;enable IRQ1 to display menu again
262         110E 7FDC 7260                      bclr            #6,@SCI0_SSR            ;clear RDRF
263         1112 5670                           rte
264         1114 38A8                   EVEN:   mov.b           R0L,@DADR0              ;put hex value into channel 0 data register
265         1116 7FAA 7060                      bset            #6,@DACR                ;start D/A conversion at channel 0
266         111A 40EE                           bra             END_CONV
267         111C 0B85                   FOUR:   adds            #2,R5                   ;point to location of second digit
268         111E 6850                           mov.b           @R5,R0H
269         1120 A01F                           cmp.b           #H'1F,R0H               ;is second digit a 1?
270         1122 4712                           beq             FOUR_1                  ;if yes, go to FOUR_1
271         1124 A022                           cmp.b           #H'22,R0H               ;is second digit a 4?
272         1126 470A                           beq             FOUR_4                  ;if yes, go to FOUR_4
273         1128 A025                           cmp.b           #H'25,R0H               ;is second digit a 7?
274         112A 4702                           beq             FOUR_7                  ;if yes, go to FOUR_7
275         112C 40B8                           bra             NONE                    ;if none, go to NONE
276         112E F8EF                   FOUR_7: mov.b           #H'EF,R0L               ;second digit is a 7
277         1130 40CE                           bra             TEST
278         1132 F8DF                   FOUR_4: mov.b           #H'DF,R0L               ;second digit is a 4
279         1134 40CA                           bra             TEST
280         1136 F8CF                   FOUR_1: mov.b           #H'CF,R0L               ;second digit is a 1
281         1138 40C6                           bra             TEST
282                                     BR_TEST:
283         113A 40C4                           bra             TEST

284                                     BRANCH_0:
285         113C 4062                           bra             ZERO
286         113E 4042                   BR_ONE: bra             ONE
287         1140 4020                   BR_TWO: bra             TWO
288         1142 0B85                   THREE:  adds            #2,R5                   ;point to location of second digit
289         1144 6850                           mov.b           @R5,R0H
290         1146 A01F                           cmp.b           #H'1F,R0H               ;is second digit a 1?
291         1148 4712                           beq             THREE_1                 ;if yes, go to THREE_1
292         114A A022                           cmp.b           #H'22,R0H               ;is second digit a 4?
293         114C 470A                           beq             THREE_4                 ;if yes, go to THREE_4
294         114E A026                           cmp.b           #H'26,R0H               ;is second digit an 8?
295         1150 4702                           beq             THREE_8                 ;if yes, go to THREE_8
296         1152 4092                           bra             NONE                    ;if none, go to NONE
297                                     THREE_8:
298         1154 F8BF                           mov.b           #H'BF,R0L               ;second digit is an 8
299         1156 40A8                           bra             TEST
300                                     THREE_4:
301         1158 F8AF                           mov.b           #H'AF,R0L               ;second digit is a 4
302         115A 40A4                           bra TEST
303                                     THREE_1:
304         115C F89F                           mov.b           #H'9F,R0L               ;second digit is a 1
305         115E 40A0                           bra             TEST
306                                     BR_NONE:
307         1160 4084                           bra             NONE
308         1162 0B85                   TWO:    adds            #2,R5                   ;point to location of second digit
309         1164 6850                           mov.b           @R5,R0H
310         1166 A020                           cmp.b           #H'20,R0H               ;is second digit a 2?
311         1168 4712                           beq             TWO_2                   ;if yes, go to TWO_2
312         116A A023                           cmp.b           #H'23,R0H               ;is second digit a 5?
313         116C 470A                           beq             TWO_5                   ;if yes, go to TWO_5
```


HITACHI

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

```
Microtec Research ASMH83    Version 1.0A    Mar 12 11:22:35 1993     Page 5


Line        Addr
314         116E A026                           cmp.b           #H'26,R0H               ;is second digit an 8?
315         1170 4702                           beq             TWO_8                   ;if yes, go to TWO_8
316         1172 40EC                           bra             BR_NONE                 ;if none, go to NONE
317         1174 F88F                   TWO_8:  mov.b           #H'8F,R0L               ;second digit is an 8
318         1176 4088                           bra             TEST
319         1178 F87F                   TWO_5:  mov.b           #H'7F,R0L               ;second digit is a 5
320         117A 4084                           bra             TEST
321         117C F86F                   TWO_2:  mov.b           #H'6F,R0L               ;second digit is a 2
322         117E 4080                           bra             TEST
323                                     BR1_TEST:
324         1180 40B8                           bra             BR_TEST
325         1182 0B85                   ONE:    adds            #2,R5                   ;point to location of second digit

326         1184 6850                           mov.b           @R5,R0H
327         1186 A021                           cmp.b           #H'21,R0H               ;is second digit a 3?
328         1188 4712                           beq             ONE_3                   ;if yes, go to ONE_3
329         118A A024                           cmp.b           #H'24,R0H               ;is second digit a 6?
330         118C 470A                           beq             ONE_6                   ;if yes, go to ONE_6
331         118E A027                           cmp.b           #H'27,R0H               ;is second digit a 9?
332         1190 4702                           beq             ONE_9                   ;if yes, go to ONE_9
333         1192 40CC                           bra             BR_NONE                 ;if none, go to NONE
334         1194 F85F                   ONE_9:  mov.b           #H'5F,R0L               ;second digit is a 9
335         1196 40A2                           bra             BR_TEST
336         1198 F84F                   ONE_6:  mov.b           #H'4F,R0L               ;second digit is a 6
337         119A 409E                           bra             BR_TEST
338         119C F83F                   ONE_3:  mov.b           #H'3F,R0L               ;second digit is a 3
339         119E 409A                           bra             BR_TEST
340         11A0 0B85                   ZERO:   adds            #2,R5                   ;point to location of second digit
341         11A2 6850                           mov.b           @R5,R0H
342         11A4 A021                           cmp.b           #H'21,R0H               ;is second digit a 3?
343         11A6 4712                           beq             ZERO_3                  ;if so, go to ZERO_3
344         11A8 A024                           cmp.b           #H'24,R0H               ;is second digit a 6?
345         11AA 470A                           beq             ZERO_6                  ;if so, go to ZERO_6
346         11AC A027                           cmp.b           #H'27,R0H               ;is second digit a 9?
347         11AE 4702                           beq             ZERO_9                  ;if yes, go to ZERO_9
348         11B0 40AE                           bra             BR_NONE                 ;if none, go to NONE
349         11B2 F82F                   ZERO_9: mov.b           #H'2F,R0L               ;second digit is a 9
350         11B4 4084                           bra             BR_TEST
351         11B6 F81F                   ZERO_6: mov.b           #H'1F,R0L               ;second digit is a 6
352         11B8 4080                           bra             BR_TEST
353         11BA F80F                   ZERO_3: mov.b           #H'0F,R0L               ;second digit is a 3
354         11BC 40C2                           bra             BR1_TEST
355                                             .end
```