Interfacing A Serial EEPROM to the National HPC16083

National Semiconductor
Application Note 552
Brian Marley
September 1988

ABSTRACT

This application note describes how to interface the HPC16083 High-Performance microController to a MICROWIRE™ serial EEPROM (Electrically Erasable Programmable Read-Only Memory) device. The technique uses interrupt-driven scheduling from one of the eight on-chip timers, and so can run in the ''background'', sharing the HPC gracefully with other control applications running at the same time. Source code is included.

1.0 INTRODUCTION

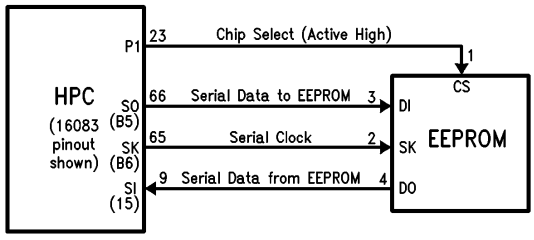
It is often the case in control-oriented applications that a piece of equipment, on being installed, must be set up with certain semi-permanent configuration mode settings. In the past, jumpers and switches have been the methods used, but in recent years these have been largely supplanted by EEPROM devices, which hold more information and are not prone to mechanical problems. In addition, the presence of an EEPROM allows certain information about the status of the equipment (for example, in printers, a page or character count for monitoring the ''age'' of the cartridge or print head) to be stored to assist in maintenance.

The most cost-effective type of EEPROM device is one with a serial interface, such as the 256-bit NMC9306 (COP494) or the 1024-bit NMC9345 (COP495). These reside in an 8-pin DIP package, and require only four connections (besides V_{CC} and Ground). These connections are provided by the HPC family of High-Performance Microcontrollers, on a serial port called the MICROWIRE/PLUS™ Interface.

Because one of the HPC's strong suits is Concurrent Control applications (applications in which several control tasks are executing simultaneously, scheduled by interrupts), the code given in this exercise is written to be completely interrupt-driven as well. Instead of timing events with software loops, interrupts from HPC Timer T5 are used both to signal the end of each MICROWIRE transfer and to time the ERASE and WRITE pulse durations for the EEPROM.

2.0 CONNECTIONS AND COMMANDS

The connection between the HPC and the EEPROM device is a completely traditional MICROWIRE connection, as shown in *Figure 1*. The SI (Serial Input), SO (Serial Output) and SK (Serial Clock) signals of the HPC connect directly to the DO, DI and SK pins of the EEPROM, respectively. The EEPROM's required Chip Select signal (CS: active high) could come from any port bit of the HPC, but the P1 pin of Port P was chosen because Port P pins present zeroes on reset (instead of floating), and this will automatically deselect the EEPROM.



TL/DD/9978–1

FIGURE 1. MICROWIRE/PLUS Connections

To communicate with the EEPROM, the signal CS (pin P1) is set high, and then each 8-bit serial transfer is triggered by writing a value to the HPC's eight-bit SIO register, which is effectively just a shift register. The data placed into the SIO register is shifted out, most-significant bit first, and eight clock pulses are presented on the SK pin corresponding to each shift. Serial data is simultaneously accepted from the SI pin, and at the end of the eight clock pulses the SIO register has been changed to reflect the value presented by the EEPROM (if any). The timing involved in a single MICROWIRE transfer is shown in *Figure 2*.

While reading from the EEPROM, the value written to SIO doesn't matter, since it is ignored by the EEPROM. The CS signal must be active throughout a command (which may involve more than one eight-bit transfer), and it must be set inactive between commands for at least one microsecond. Also, the time between an ERASE or WRITE command and the following command (as measured by the amount of time the CS signal remains low between them) determines the length of the corresponding ERASE or WRITE pulse within the EEPROM chip. These pulse widths have strict limits which, if exceeded, can damage some EEPROMs.

EEPROM commands are 8-bit values. However, they must start with an additional ''1'' bit (the Start bit), and READ commands require a trailing ''pad'' bit, to provide timing control for the access. Since HPC MICROWIRE transfers must consist of integral numbers of 8-bit transfers, at least two such transfers must be used per command.

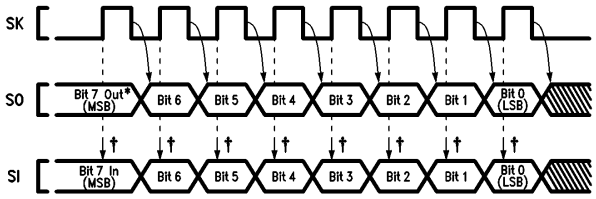
Note that the formats shown below (with 6 address bits) support an EEPROM with up to 1K bits (64 16-bit words). To use a 256-bit EEPROM, one would not specify an address greater than binary 001111, because the two most-significant address bits are ignored by the EEPROM.

2.1 Read Commands

Reading a 16-bit word from the EEPROM is accomplished with a single READ command. For the READ command, the format is:

```
      0 0 0 0 0 0 1 1  0 A A A A A A 0
      |- - - - -| |                  |
           |      start bit          pad bit
  leading zeroes
    (ignored)
```

where the bits marked ''A'' constitute the address of the EEPROM word to be accessed. These two command transfers are followed by two additional 8-bit transfers, in which the 16 bits of data from the addressed EEPROM word are read by the HPC (most significant bit first).



TL/DD/9978–2

*This bit becomes valid immediately when the transmitting device loads its SIO register. The HPC guarantees it to be valid for at least 1 full SK period before the rising edge of the first SK pulse presented.

† Arrows indicate points at which SI is sampled.

FIGURE 2. MICROWIRE/PLUS Transfer

Master presents eight pulses on SK pin; each pulse transfers one bit in and out.

2.2 Write Commands

To write data into the EEPROM, a sequence of commands is entered:

```
an EWEN command (Erase/Write Enable):
        0 0 0 0 0 0 0 1     0 0 1 1 0 0 0 0
an ERASE command:
        0 0 0 0 0 0 0 1     1 1 A A A A A A
              ("A" = Address bits,
                 most-significant bit first)
a pause of 16 to 25 milliseconds, with CS
low,
a WRITE command:
        0 0 0 0 0 0 0 1     0 1 A A A A A A
        D D D D D D D D     D D D D D D D D
("A" = Address bits,
              "D" = Data bits,
                 most-significant bit first)
a pause of 16 to 25 milliseconds, with CS
low,
and, finally, an EWDS command (Erase/Write
Disable):
        0 0 0 0 0 0 0 1     0 0 0 0 0 0 0 0
```

3.0 LISTING AND COMMENTARY

The listing provided shows three necessary segments of a program to access the EEPROM device:

1) initialization of the MICROWIRE/PLUS port on the HPC,
2) two program fragments of a Main Program which would initiate a Read or a Write operation,
3) an interrupt service routine (attached to Timer T5) which actually performs the transfers.

3.1 Initialization

On receiving a Reset signal, the HPC begins execution at the label ''start''. It loads the PSW register (to select 1 Wait state), and then removes all interrupt enables.

At label ''sram'', all RAM within the HPC is initialized to zero.

At ''suwire'', the MICROWIRE/PLUS interface pins are initialized. The MICROWIRE/PLUS interface is then set to the CKI/128 bit rate (125 KHz clocking at 16 MHz crystal frequency). The internal interface is not completely cleared by the Reset signal, so the firmware must set it up and wait (at label ''suwlp'') for the interface to become ready. Once this has been done, a byte of all zeroes is sent to the EEPROM to terminate any Write operation that might have been in progress when the Reset was received.

At ''tminit'', the timers T1–T7 are stopped and any interrupts pending from timers T0–T7 are cleared. The individual timer interrupt enables are then cleared.

The program then continues to label ''minit'', which initializes the variables in the HPC's on-chip RAM to their proper contents.

At label ''runsys'', the necessary interrupt is enabled (from the timers), and execution continues to the body of the Main Program.

There follow now two fragments of illustrative main program code which can be used to trigger the process of reading and writing the EEPROM.

3.2 Reading

The main program and interrupt routines given here enable reading from one to eight bytes from the EEPROM, starting at the beginning of any word.

At label ''rnvr'', an EEPROM READ command is constructed from the EEPROM starting address and placed in the variable ''nvrcmd''. The number of bytes to be transferred is placed in the variable ''nvrnum''. Control is then transferred to the label ''nvrx'', where Timer T5 is set up to generate scheduling interrupts for reading data from the EEPROM.

The variable ''nvrs'' indicates the state of an EEPROM access from one interrupt to another: its top bit (''nvravl'') shows whether the EEPROM is already being used, bit 6 (''nvrwr'') shows whether it is being written or read, and the low-order 4 bits hold a state number, which is used to transfer control to the appropriate code within the Timer T5 interrupt service routine.

On each Timer T5 interrupt (see labels ''tmrint'', ''t5poll'', ''t5int''), the timer is stopped, a check is made to determine whether the EEPROM is being read or written (T5 interrupts are used for both), and then a multiway branch (jidw) is performed based on the state number in the variable ''nvrs''. The state number is incremented on each interrupt. On a Read transfer, five states are entered, at the following labels:

t5rd0 activates the chip select to the EEPROM and initiates the MICROWIRE transfer to send the first byte of a READ command. Timer T5 is started to time out the MICROWIRE transfer.

t5rd1 sends the second byte of the READ command. Timer T5 is started to time out the MICROWIRE transfer.

t5rd2 initiates the MICROWIRE transfer to read the first byte of data from the current EEPROM word. Timer T5 is started to time out the MICROWIRE transfer.

t5rd3 accepts the first byte of the data into the high-order byte of the variable ''nvword'', and initiates the transfer to read the second byte of the current EEPROM word. Timer T5 is started to time out the MICROWIRE transfer.

t5rd4 accepts the second byte from the EEPROM into the low-order byte of the variable ''nvword'', and then moves the word into the EEPROM string buffer, called ''nvrbuf'', using a pointer called ''nvrptr''. It then checks whether the requested number of bytes has been read (by decrementing the ''nvrnum'' variable). If so, it leaves Timer T5 stopped, disables its interrupt and returns. This would also be the proper place to set a semaphore flag to acknowledge to the main program that the reading is complete. (Code for this is not included here; it would vary from system to system.) If the requested number of bytes has not yet been read, it increments the address field of the READ command in ''nvrcmd'', resets the state field in ''nvrs'' to zero, leaves Timer T5 interrupts enabled, and jumps directly to the ''t5rd0'' routine to continue.

3.3 Writing

At label ''wnvr'', an EEPROM ERASE command is constructed from the word address supplied by the CPU. The 16-bit value to be written is placed in the variable ''nvword''. As in the READ-NVR command above, the ''nvrs'' variable is initialized to select the first state of an EEPROM write operation, and Timer T5 is used to provide the interrupts that schedule the steps. There are 13 states involved in writing a word to the EEPROM, at the following labels:

t5wr0 activates the chip select signal to the EEPROM, and sends the first byte of an EWEN command to enable ERASE and WRITE commands. Timer T5 is started to time out the MICROWIRE transfer.

t5wr1 sends the second byte of the EWEN command. Timer T5 is started to time out the MICROWIRE transfer.

t5wr2 removes the chip select signal briefly (to signal the beginning of a new command), then sends the first byte of an ERASE command. Timer T5 is started to time out the MICROWIRE transfer.

t5wr3 sends the second byte of the ERASE command, from the variable ''nvrcmd''. Timer T5 is started to time out the MICROWIRE transfer.

t5wr4 removes the chip select signal, then sets up the Timer T5 interval to 20 milliseconds, to time the duration of the EEPROM's internal Erase pulse.

t5wr5 (entered 20 milliseconds after ''t5wr4'') re-asserts the chip select signal to the EEPROM, and transfers the first byte of a WRITE command. Timer T5 is started to time out the MICROWIRE transfer.

t5wr6 alters the command in ''nvrcmd'' to a WRITE command, then transfers it as the second command byte to the EEPROM. Timer T5 is started to time out the MICROWIRE transfer.

t5wr7 transfers the first byte of data to be written. Timer T5 is started to time out the MICROWIRE transfer.

t5wr8 transfers the second byte of data to be written. Timer T5 is started to time out the MICROWIRE transfer.

t5wr9 removes the chip select signal, then sets up the Timer T5 interval to 20 milliseconds, to time the duration of the EEPROM's internal Write pulse.

t5wr10 (entered 20 milliseconds after ''t5wr9'') re-asserts the chip select signal to the EEPROM, and transfers the first byte of an EWDS command (Erase/Write Disable). Timer T5 is started to time out the MICROWIRE transfer.

t5wr11 transfers the second byte of the EWDS command. Timer T5 is started to time out the MICROWIRE transfer.

t5wr12 removes the chip select signal to the EEPROM, keeps Timer T5 stopped, disables its interrupt, and returns. This would also be the proper place to set a semaphore flag to acknowledge to the main program that the writing is complete. (Code for this is not included here; it would vary from system to system.)

3.4 Source Listing

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)          EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                             PAGE   1


     1                                  .title  EEPROM,'HPC-Based Driver for NMC9306/9345'
     2
     3                          ; This code is written to drive either the 256-bit NMC9306 (COP494)
     4                          ;   or the 1024-bit NMC9345 (COP495) MICROWIRE(tm) EEPROM.
     5
     6                          ; NOTE:  Timing values assume that the HPC is running at 16MHz
     7                          ;          crystal frequency.  For correct programming pulse
     8                          ;          widths, one should not deviate far from this without
     9                          ;          adjusting the timing constant below.
    10
    11 4E1F                     TIMCON  =       19999   ; 20000 counts at 1 usec = 20 msec.
    12                                                  ;  Timing constant for ERASE and WRITE
    13                                                  ;  pulse widths.
    14
```

TL/DD/9978–3

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)          EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                             PAGE   2
Declarations:  Register Addresses

    15                                  .form   'Declarations:  Register Addresses'
    16
    17 00C0                     psw     =       x'C0:w  ; PSW register
    18 00C8                     al      =       x'C8:b  ; Low byte of Accumulator.
    19 00C9                     ah      =       x'C9:b  ; High byte of Accumulator.
    20 00CC                     bl      =       x'CC:b  ; Low byte of Register B.
    21 00CD                     bh      =       x'CD:b  ; High byte of Register B.
    22 00CE                     xl      =       x'CE:b  ; Low byte of Register X.
    23 00CF                     xh      =       x'CF:b  ; High Byte of Register X.
    24
    25 00D0                     enir    =       x'D0:b
    26 00D2                     irpd    =       x'D2:b
    27 00D4                     ircd    =       x'D4:b
    28 00D6                     sio     =       x'D6:b
    29 00D8                     porti   =       x'D8:b
    30 00E0                     obuf    =       x'E0:b  ; (Low byte of PORTA.)
    31 00E1                     portah  =       x'E1:b  ; High byte of PORTA.
    32 00E2                     portb   =       x'E2:w
    33 00E2                     portbl  =       x'E2:b  ; Low byte of PORTB.
    34 00E3                     portbh  =       x'E3:b  ; High byte of PORTB.
    35 00E6                     upic    =       x'E6:b
    36 00F0                     ibuf    =       x'F0:b  ; (Low byte of DIRA.)
    37 00F1                     dirah   =       x'F1:b  ; High byte of DIRA.
    38 00F2                     dirb    =       x'F2:w
    39 00F2                     dirbl   =       x'F2:b  ; Low byte of DIRB.
    40 00F3                     dirbh   =       x'F3:b  ; High byte of DIRB.
    41 00F4                     bfun    =       x'F4:w
    42 00F4                     bfunl   =       x'F4:b  ; Low byte of BFUN.
    43 00F5                     bfunh   =       x'F5:b  ; High byte of BFUN.
    44
    45 0104                     portd   =       x'0104:b
    46 0120                     enu     =       x'0120:b
    47 0122                     enui    =       x'0122:b
    48 0124                     rbuf    =       x'0124:b
    49 0126                     tbuf    =       x'0126:b
    50 0128                     enur    =       x'0128:b
    51
    52 0140                     t4      =       x'0140:w
    53 0142                     r4      =       x'0142:w
    54 0144                     t5      =       x'0144:w
```

TL/DD/9978–4

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)            EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                                PAGE   3
Declarations:  Register Addresses

    55 0146                     r5      =       x'0146:w
    56 0148                     t6      =       x'0148:w
    57 014A                     r6      =       x'014A:w
    58 014C                     t7      =       x'014C:w
    59 014E                     r7      =       x'014E:w
    60 0150                     pwmode  =       x'0150:w
    61 0150                     pwmdl   =       x'0150:b  ; Low byte of PWMODE.
    62 0151                     pwmdh   =       x'0151:b  ; High byte of PWMODE.
    63 0152                     portp   =       x'0152:w
    64 0152                     portpl  =       x'0152:b  ; Low byte of PORTP.
    65 0153                     portph  =       x'0153:b  ; High byte of PORTP.
    66 015C                     eicon   =       x'015C:w
    67
    68 0182                     t1      =       x'0182:w
    69 0184                     r1      =       x'0184:w
    70 0186                     r2      =       x'0186:w
    71 0188                     t2      =       x'0188:w
    72 018A                     r3      =       x'018A:w
    73 018C                     t3      =       x'018C:w
    74 018E                     divby   =       x'018E:w
    75 018E                     divbyl  =       x'018E:b  ; Low byte of DIVBY.
    76 018F                     divbyh  =       x'018F:b  ; High byte of DIVBY.
    77 0190                     tmmode  =       x'0190:w
    78 0190                     tmmdl   =       x'0190:b  ; Low byte of TMMODE.
    79 0191                     tmmdh   =       x'0191:b  ; High byte of TMMODE.
    80 0192                     t0con   =       x'0192:b
    81
    82
```

TL/DD/9978–5

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)            EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                                PAGE   4
Declarations:  Bit Positions

    83                                  .form   'Declarations:  Bit Positions'
    84
    85                          ; Name          Position        Register(s)
    86                          ; ----          --------        -----------
    87
    88 0000                     gie     =       0       ; enir
    89 0000                     i0      =       0       ; porti only
    90 0002                     i2      =       2       ; enir, irpd, ircd
    91 0003                     i3      =       3       ; enir, irpd, ircd
    92 0004                     i4      =       4       ; enir, irpd, ircd
    93 0005                     tmrs    =       5       ; enir, irpd
    94 0006                     uart    =       6       ; enir, irpd
    95 0007                     ei      =       7       ; enir, irpd
    96
    97 0001                     uwmode  =       1       ; ircd
    98 0000                     uwdone  =       0       ; irpd
    99
   100 0000                     tbmt    =       0       ; enu
   101 0001                     rbfl    =       1       ; enu
   102 0004                     b8or9   =       4       ; enu
   103 0005                     xbit9   =       5       ; enu
   104 0002                     wakeup  =       2       ; enur
   105 0003                     rbit9   =       3       ; enur
   106 0006                     frmerr  =       6       ; enur
   107 0007                     doeerr  =       7       ; enur
   108 0000                     eti     =       0       ; enui
   109 0001                     eri     =       1       ; enui
   110 0002                     xtclk   =       2       ; enui
   111 0003                     xrclk   =       3       ; enui
   112 0007                     b2stp   =       7       ; enui
   113
   114 0000                     wrrdy   =       0       ; upic
   115 0001                     rdrdy   =       1       ; upic
   116 0002                     la0     =       2       ; upic
   117 0003                     upien   =       3       ; upic
   118 0004                     b8or16  =       4       ; upic
   119
   120 0000                     t0tie   =       0       ; tmmdl
   121 0001                     t0pnd   =       1       ; tmmdl
   122 0003                     t0ack   =       3       ; tmmdl
```

TL/DD/9978–6

```
123 0004                    t1tie   =       4       ; tmmdl
124 0005                    t1pnd   =       5       ; tmmdl
125 0006                    t1stp   =       6       ; tmmdl
126 0007                    t1ack   =       7       ; tmmdl
127 0000                    t2tie   =       0       ; tmmdh
128 0001                    t2pnd   =       1       ; tmmdh
129 0002                    t2stp   =       2       ; tmmdh
130 0003                    t2ack   =       3       ; tmmdh
131 0004                    t3tie   =       4       ; tmmdh
132 0005                    t3pnd   =       5       ; tmmdh
133 0006                    t3stp   =       6       ; tmmdh
134 0007                    t3ack   =       7       ; tmmdh
135
136 0000                    t4tie   =       0       ; pwmdl
137 0001                    t4pnd   =       1       ; pwmdl
138 0002                    t4stp   =       2       ; pwmdl
139 0003                    t4ack   =       3       ; pwmdl
140 0004                    t5tie   =       4       ; pwmdl
141 0005                    t5pnd   =       5       ; pwmdl
142 0006                    t5stp   =       6       ; pwmdl
143 0007                    t5ack   =       7       ; pwmdl
144 0000                    t6tie   =       0       ; pwmdh
145 0001                    t6pnd   =       1       ; pwmdh
146 0002                    t6stp   =       2       ; pwmdh
147 0003                    t6ack   =       3       ; pwmdh
148 0004                    t7tie   =       4       ; pwmdh
149 0005                    t7pnd   =       5       ; pwmdh
150 0006                    t7stp   =       6       ; pwmdh
151 0007                    t7ack   =       7       ; pwmdh
152
153 0000                    t4out   =       0       ; portpl
154 0003                    t4tfn   =       3       ; portpl
155 0004                    t5out   =       4       ; portpl
156 0007                    t5tfn   =       7       ; portpl
157 0000                    t6out   =       0       ; portph
158 0003                    t6tfn   =       3       ; portph
159 0004                    t7out   =       4       ; portph
160 0007                    t7tfn   =       7       ; portph
161
162 0000                    eipol   =       0       ; eicon
```



```
163 0001                    eimode  =       1       ; eicon
164 0002                    eiack   =       2       ; eicon
165
166 0000                    txd     =       0       ; portbl, dirbl, bfunl
167 0003                    t2in    =       3       ; portbl, dirbl
168 0005                    so      =       5       ; portbl, dirbl, bfunl
169 0006                    sk      =       6       ; portbl, dirbl, bfunl
170
171
```

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)           EEPROM                                    03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                                PAGE   7
Space Declarations

                                .form   'Space Declarations'
0000                            .sect   DSECT,BASE,REL

                        ;WORD-ALIGNED VARIABLES

0000                    stackb: .dsw    16      ; Space for 16 words.
0020                    nvrbuf: .dsw    4       ; EEPROM String Buffer.
0028                    nvrptr: .dsw 1  ; Pointer into EEPROM Data buffer.
002A                    nvword: .dsw 1  ; Scratch location for gathering EEPROM data as words.

                        ;BYTE-ALIGNED VARIABLES

002C                    nvrcmd: .dsb 1  ; Current EEPROM command.
002D                    nvrnum: .dsb 1  ; Byte count for current EEPROM Read command.
002E                    nvrs:   .dsb 1  ; EEPROM status byte:  phase number for sequencing MICROWIRE
                                        ;  transfers.

                        ;BIT DEFINITIONS

                                ; NVRS byte:  Status of EEPROM MICROWIRE transfers.
                                ;               Contains phase (step number) of current EEPROM command
                                ;               in low-order 4 bits.  Top two bits are as follows:
0007                    nvravl= 7       ; When set, indicates that no EEPROM command is in progress.
0006                    nvrwr=  6       ; 0 means an EEPROM Read is in progress;  1 means EEPROM Write.


                                                                                  TL/DD/9978-9

NSC ASMHPC, Version E2 (Nov 02 15:51 1987)           EEPROM                                    03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                                PAGE   8
Code Section

                                .form   'Code Section'
0000                            .sect   CSECT,ROM16,REL ; Code space.

0000 B70008C0           start:  ld      psw,#x'08       ; Set one WAIT state.
0004 9700D0                     ld      enir,#x'00      ; Disable interrupts
                                                        ;  individually.

0007                    sram:                   ; Clear all RAM locations.
                                                ; Basepage bank:
0007 8D00BE                     ld      BK,#x'0000,#x'00BE      ; Establish loop base and limit.
000A 00                 sraml1: clr     A
000B E1                         xs      A,[B+].w
000C 62                         jp      sraml1

                                        ; Non-basepage bank:
000D A701C001FE                 ld      BK,#x'01C0,#x'01FE      ; Establish loop base and limit.
0012 00                 sraml2: clr     A
0013 E1                         xs      A,[B+].w
0014 62                         jp      sraml2

0015                    suwire:                         ; MICROWIRE setup.
                                                        ; (EEPROM is automatically
                                                        ;  deselected on reset, since
                                                        ;  Port P is cleared.)

0015 96F40D                     sbit    so,bfunl        ; Enable SO output.
0018 96F20D                     sbit    so,dirbl
001B 96E21E                     rbit    sk,portbl       ; Set up SK output.
001E 96F20E                     sbit    sk,dirbl
0021 96F40E                     sbit    sk,bfunl
0024 96D409                     sbit    uwmode,ircd     ; Set Master Mode.
0027 87222518EAB                ld      divby,#x'2225   ; Set MICROWIRE frequency.

002D 96D210             suwlp:  ifbit   uwdone,irpd     ; Wait until MICROWIRE
0030 41                         jp      snvr1           ;  interface ready (uWDONE
0031 64                         jp      suwlp           ;  bit set).

0032                    snvr1:                          ; Cancel any EEPROM Write in progress:
0032 B601520C                   sbit    t5out,portpl    ; Set EEPROM Chip Select active.
0036 9700D6                     ld      sio,#0          ; Send a byte of zeroes.

                                                                                  TL/DD/9978-10
```

Code Section

```
238 0039 96D210             suwlp1: ifbit   uwdone,irpd     ; Wait until MICROWIRE
239 003C 41                         jp      snvr2           ;  interface ready (uWDONE
240 003D 64                         jp      suwlp1          ;  bit set).
241 003E B6015210           snvr2:  rbit    t5out,portpl    ; Remove EEPROM Chip Select.
242
243 0042 83080192B          tminit: ld      t0con,#x'08
244 0047 87444001900AB              ld      tmmode,#x'4440  ; Stop timers T1, T2, T3.
245 004D 83550180AB                 ld      divby,#x'0055   ; MICROWIRE frequency set
246                                                         ;  to CKI/128.
247 0052 87CCC8019000AB             ld      tmmode,#x'CCC8  ; Clear and disable timer
248                                                         ;  T0-T3 interrupts.
249
250 0058 87444401500AB              ld      pwmode,#x'4444  ; Stop timers T4-T7.
251 005E 40                         nop                     ; Wait for Pending bits to
252 005F 40                         nop                     ;  trickle through before clearing them.
253 0060 87CCCC01500AB              ld      pwmode,#x'CCCC  ; Clear and disable
254                                                         ;  interrupts from all
255                                                         ;  PWM timers.
256
257 0066 87FFFF0146AB               ld      r5,#x'FFFF      ; No modulus for EEPROM timer.
258
```

TL/DD/9978–11

Main Program Initialization

```
259                                 .form   'Main Program Initialization'
260
261 006C                    minit:
262 006C 97802E          R          ld      nvrs,#x'80      ; Set EEPROM available.
263 006F B7002028        R          ld      nvrptr,#nvrbuf  ; Set EEPROM pointer to start of buffer.
264
265 0073                    runsys:                 ; Enable timer interrupts, and go to main.
266
267 0073 96D00D                     sbit    tmrs,enir       ; Enable timer interrupts.  (Done here
268                                                         ;  to allow engine commands without an
269                                                         ;  INITIALIZE command first.)
270 0076 96D008                     sbit    gie,enir        ; Enable interrupt system.
271
272
```

TL/DD/9978–12

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)          EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                           PAGE  11
Main Program Fragments

                                    .form   'Main Program Fragments'

                            ;  These values are declared as constants;  more typically they would be
                            ;     contained within variables.  Note that the pound-sign character must
                            ;     then be deleted in the instructions referencing them.

0000                        nvradr  =       0       ; EEPROM address: change to suit your application.
ABCD                        nvrdta  =       x'ABCD  ; Written data:  change to suit.
0004                        nvrbyt  =       4       ; Number of bytes to read (1-8):  change to suit.

                            ; Read Fragment:  reads up to 4 words (8 bytes) from EEPROM.

0079 9000                   rnvr:   ld      A,#nvradr       ; Get NVR starting address.
007B 993F                           and     A,#x'3F         ; Truncate to legal limit.
007D E7                             shl     A               ; Create NVR READ command.
007E 8B2C             R             st      A,nvrcmd        ; Place it in memory.
0080 9004                           ld      A,#nvrbyt       ; Get number of bytes requested.
0082 8B2D             R             st      A,nvrnum        ; Save byte count in memory.
0084 97002E           R             ld      nvrs,#0         ; Set up NVR access status flags:
                                                            ;  Read transfer in progress, first phase.
0087 B7002028         R             ld      nvrptr,#nvrbuf  ; Reset buffer pointer to beginning.
008B 4E                             jmpl    nvrx            ; Go start up transfer.


                            ; Write Fragment:  writes one word to EEPROM.

008C B7ABCD2A         R     wnvr:   ld      nvword,#nvrdta  ; Get data word.
0090 9000                           ld      A,#nvradr       ; Get EEPROM address.
0092 993F                           and     A,#x'3F         ; Mask it for proper range.
0094 8B2C             R             st      A,nvrcmd        ; Store it in Command byte in memory.
                                                            ;  (Opcode = 00 at this point.)
0096 97402E           R             ld      nvrs,#x'40      ; Set up NVR access status flags:
                                                            ;  Write transfer in progress, first phase.
0099 40                             jmpl    nvrx            ; Go start up transfer.


                            ; Common routine, performed by both READ and WRITE.

009A                        nvrx:                           ; Start interrupts from Timer T5 to schedule
                                                            ;  accesses to EEPROM.
```

TL/DD/9978–13

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)          EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                           PAGE  12
Main Program Fragments

009A 87FFFF0146AB                   ld      r5,#x'FFFF      ; Interrupts are not repetitive;  give R5 a
                                                            ;  high value.
00A0 83000144AB                     ld      t5,#0           ; Set Timer T5 to interrupt (almost)
                                                            ;  immediately when started.
00A5 B601500C                       sbit    t5tie,pwmdl     ; Enable interrupt from Timer T5.
00A9 B601501E                       rbit    t5stp,pwmdl     ; Start Timer T5.

                            ; *** One could replace the following instruction with one that
                            ; ***  looks for an appropriate semaphore bit to be set, indicating
                            ; ***  that the requested operation has been completed.  See other
                            ; ***  comments beginning with "****".

00AD 60                             jp      .               ; Stops HPC, except for interrupt service.


                            ;       END OF MAIN PROGRAM FRAGMENTS.
```

TL/DD/9978–14

```
                                        .form   'Timer Interrupt Handler'

                                ;       The Timer T5 interrupt service routine does all the work.  Each
                                ;          interrupt sequences the next step of the READ or WRITE
                                ;          operation in progress.

FFF4 AE00               R               .ipt    5,tmrint        ; Declare entry point for Timer Interrupt.

00AE AFC8                       tmrint: push    A               ; Save context.
00B0 AFC0                               push    psw.w           ;

00B2 B601501S                   t5poll: ifbit   t5pnd,pwmdl     ; Poll for Timer T5 interrupt (EEPROM Timing
00B6 41                                 jmpl    t5int           ;  Interrupt).

00B7 60                                 jp      .               ; Otherwise, error.  Stop HPC.

00B8 B601500E                   t5int:  sbit    t5stp,pwmdl     ; Stop Timer T5.
00BC B601500F                           sbit    t5ack,pwmdl     ; Clear interrupt request.  (Doing this
                                                                ;  immediately is acceptable here.)
00C0 962E16             R               ifbit   nvrwr,nvrs      ; Check whether Read or Write operation is
                                                                ;  is in progress.
00C3 9483                               jmpl    t5wr            ; If Write, go perform
                                                                ;  Enable/Erase/Write/Disable operation.
00C5                            t5rd:                           ; Else, program is reading from EEPROM.
00C5 882E               R               ld      A,nvrs          ; Get phase info.
00C7 892E               R               inc     nvrs            ; Increment memory value for next T5 interrupt.
00C9 990F                               and     A,#x'0F         ; Extract phase number.
00CB E7                                 shl     A               ; Jump based on this number.
00CC 40
00CD                                    .odd
00CD EC                                 jidw
00CE 0A00                               .ptw    t5rd0,t5rd1,t5rd2,t5rd3,t5rd4
00D0 1B00
00D2 2800
00D4 3500
00D6 4500

00D8 B601520C                   t5rd0:  sbit    t5out,portpl    ; Set chip select signal to EEPROM.
00DC 9703D6                             ld      sio,#x'03       ; Send first part of NVR Read command.
                                                                ;  Format is:  1/10/A5-A0/0 ,
```



```
                                                                ;   where first bit is start bit (always '1'),
                                                                ;         next two bits are operation (10=read),
                                                                ;         next 6 bits are EEPROM address,
                                                                ;         last bit is "padding" for access time.
                                                                ;  This phase sends top two bits of command.
00DF 835A0144AB                         ld      t5,#90          ; Set up for interrupt after MICROWIRE transfer.
00E4 B601501E                           rbit    t5stp,pwmdl     ; Start Timer T5.
00E8 B40151                             jmpl    tmrret          ; Return from interrupt.

00EB 8C2CD6             R       t5rd1:  ld      sio,nvrcmd      ; Send second part of NVR Read command (bottom
                                                                ;  eight bits).
00EE 835A0144AB                         ld      t5,#90          ; Set up for interrupt after MICROWIRE transfer.
00F3 B601501E                           rbit    t5stp,pwmdl     ; Start Timer T5.
00F7 B40142                             jmpl    tmrret          ; Return from interrupt.

00FA 9700D6                     t5rd2:  ld      sio,#0          ; Start reading MSB of EEPROM data.
00FD 835A0144AB                         ld      t5,#90          ; Set up for interrupt after MICROWIRE transfer.
0102 B601501E                           rbit    t5stp,pwmdl     ; Start Timer T5.
0106 B40133                             jmpl    tmrret          ; Return from interrupt.

0109 8CD62B             R       t5rd3:  ld      nvword+1.b,sio  ; Accept MSB of EEPROM data to word buffer.
010C 9700D6                             ld      sio,#0          ; Start reading LSB of EEPROM data.
010F 835A0144AB                         ld      t5,#90          ; Set up for interrupt after MICROWIRE transfer.
0114 B601501E                           rbit    t5stp,pwmdl     ; Start Timer T5.
0118 B40121                             jmpl    tmrret          ; Return from interrupt.

011B 8CD62A             R       t5rd4:  ld      nvword.b,sio    ; Accept LSB of EEPROM data to word buffer.
011E B601521C                           rbit    t5out,portpl    ; Remove EEPROM chip select signal.
0122 A82A               R               ld      A,nvword        ; Get EEPROM data word.
0124 AD28AB             R               st      A,[nvrptr].w    ; Store in EEPROM buffer for CPU.
0127 A928               R               inc     nvrptr          ; Increment EEPROM buffer pointer once.
0129 8A2D               R               decsz   nvrnum          ; Check whether both bytes of the word were
                                                                ;  requested.
012B 41                                 jp      t5rdh           ;  Yes: continue.
012C 45                                 jp      t5rddn          ;  No: done with reading.
012D A928               R       t5rdh:  inc     nvrptr          ; Increment EEPROM buffer pointer a second time
                                                                ;  (to signal that a whole word was input to
                                                                ;   buffer).
012F 8A2D               R               decsz   nvrnum          ; Check whether done.
0131 4A                                 jp      t5rnxt          ;  No:  Initiate another Read command.
```

```
404                                             ;  Yes:  Terminate and pass data to CPU.
405 0132 B601501C           t5rddn: rbit    t5tie,pwmdl     ; Disable Timer T5 interrupts.
406 0136 962E0F          R          sbit    nvravl,nvrs     ; Set NVR available for more commands.
407                         ;*** Here you'll want to set a semaphore bit saying that the READ
408                         ;***  transfer is done.
409 0139 B40100                    jmpl    tmrret          ; Return from interrupt.
410
411 013C                    t5rnxt:                         ; Here, more data needs to be read from the
412                                                         ;  EEPROM.  Initiate another read cycle.
413 013C 97002E          R          ld      nvrs,#x'00      ; Set up new transfer phase = 0.
414 013F 892C            R          inc     nvrcmd          ; Increment address field of NVR command.
415 0141 892C            R          inc     nvrcmd          ;  (Two increments are needed:  field starts
416                                                         ;   in Bit 1.)
417 0143 962C1F          R          rbit    7,nvrcmd        ; Prevent increments from altering operation
418                                                         ;  field.  This allows addresses to roll over.
419 0146 9581                      jmpl    t5rd            ; Rather than triggering a Timer T5 interrupt,
420                                                         ;  just jump to T5 Read interrupt service again.
421
422
423 0148                    t5wr:                   ; EEPROM Write sequence starts here.
424 0148 882E            R          ld      A,nvrs          ; Get phase info.
425 014A 892E            R          inc     nvrs            ; Increment memory value for next T5 interrupt.
426 014C 990F                       and     A,#x'0F         ; Extract phase number.
427 014E E7                         shl     A               ; Jump based on this number.
428 014F                            .odd
429 014F EC                         jidw
430 0150 1A00                       .ptw    t5wr0,t5wr1,t5wr2,t5wr3
    0152 2A00
    0154 3600
    0156 4B00
431 0158 5B00                       .ptw    t5wr4,t5wr5,t5wr6,t5wr7
    015A 6900
    015C 7900
    015E 8800
432 0160 9400                       .ptw    t5wr8,t5wr9,t5wr10,t5wr11
    0162 A000
    0164 AE00
    0166 BD00
433 0168 C800                       .ptw    t5wr12
434
```

TL/DD/9978–17

```
  435 016A B601520C          t5wr0:  sbit    t5out,portpl    ; Set chip select signal to EEPROM.
  436 016E 9701D6                    ld      sio,#x'01       ; Send start bit of EWEN command.
  437 0171 835A0144AB                ld      t5,#90          ; Set up for interrupt at end of MICROWIRE
  438                                                        ;  transfer.
  439 0176 B601501E                  rbit    t5stp,pwmdl     ; Start timer T5.
  440 017A 94C0                      jmpl    tmrret          ; Return from interrupt.
  441
  442 017C 9730D6            t5wr1:  ld      sio,#x'30       ; Send body of EWEN command.
  443 017F 835A0144AB                ld      t5,#90          ; Set up for interrupt at end of MICROWIRE
  444                                                        ;  transfer.
  445 0184 B601501E                  rbit    t5stp,pwmdl     ; Start timer T5.
  446 0188 94B2                      jmpl    tmrret          ; Return from interrupt.
  447
  448 018A B601521C          t5wr2:  rbit    t5out,portpl    ; Remove EEPROM select momentarily to signal
  449 018E 40                        nop                     ;  end of EWEN command, then:
  450 018F B601520C                  sbit    t5out,portpl
  451 0193 9701D6                    ld      sio,#x'01       ; Send Start Bit for ERASE command.
  452 0196 835A0144AB                ld      t5,#90          ; Set up for interrupt at end of MICROWIRE
  453                                                        ;  transfer.
  454 019B B601501E                  rbit    t5stp,pwmdl     ; Start timer T5.
  455 019F 949B                      jmpl    tmrret          ; Return from interrupt.
  456
  457 01A1 82C02CDA        R t5wr3:  or      nvrcmd,#x'C0    ; Change NVR Command byte to ERASE command.
  458 01A5 8C2CD6          R         ld      sio,nvrcmd      ; Send to EEPROM.
  459 01A8 835A0144AB                ld      t5,#90          ; Set up for interrupt at end of MICROWIRE
  460                                                        ;  transfer.
  461 01AD B601501E                  rbit    t5stp,pwmdl     ; Start timer T5.
  462 01B1 9489                      jmpl    tmrret          ; Return from interrupt.
  463
  464 01B3 B601521C          t5wr4:  rbit    t5out,portpl    ; Remove EEPROM chip select signal, starting
  465                                                        ;  ERASE pulse inside EEPROM.
  466 01B7 874E1F0144AB              ld      t5,#TIMCON      ; Set up for delay of 20
  467                                                        ;  milliseconds (erase pulse width).
  468 01BD B601501E                  rbit    t5stp,pwmdl     ; Start timer T5.
  469 01C1 9479                      jmpl    tmrret          ; Return from interrupt.
  470
  471 01C3 B601520C          t5wr5:  sbit    t5out,portpl    ; Set EEPROM chip select signal again, ending
  472                                                        ;  the ERASE pulse inside EEPROM.
  473 01C7 9701D6                    ld      sio,#x'01       ; Send Start bit for Write command.
  474 01CA 835A0144AB                ld      t5,#90          ; Set up for interrupt at end of MICROWIRE
```

TL/DD/9978–18

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)          EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                           PAGE  17
Timer Interrupt Handler

  475                                                       ;  transfer.
  476 01CF B601501E                   rbit    t5stp,pwmdl   ; Start timer T5.
  477 01D3 9467                       jmpl    tmrret        ; Return from interrupt.
  478
  479 01D5 962C1F          R  t5wr6:  rbit    7,nvrcmd      ; Create WRITE command in NVR Command byte.
  480 01D8 8C2CD6          R          ld      sio,nvrcmd    ; Send to EEPROM.
  481 01DB 835A0144AB                 ld      t5,#90        ; Set up for interrupt at end of MICROWIRE
  482                                                       ;  transfer.
  483 01E0 B601501E                   rbit    t5stp,pwmdl   ; Start timer T5.
  484 01E4 9456                       jmpl    tmrret        ; Return from interrupt.
  485
  486 01E6 8C2BD6          R  t5wr7:  ld      sio,nvword+1.b ; Send MSB of data to EEPROM.
  487 01E9 835A0144AB                 ld      t5,#90        ; Set up for interrupt at end of MICROWIRE
  488                                                       ;  transfer.
  489 01EE B601501E                   rbit    t5stp,pwmdl   ; Start timer T5.
  490 01F2 9448                       jmpl    tmrret        ; Return from interrupt.
  491
  492 01F4 8C2AD6          R  t5wr8:  ld      sio,nvword.b  ; Send LSB of data to EEPROM.
  493 01F7 835A0144AB                 ld      t5,#90        ; Set up for interrupt at end of MICROWIRE
  494                                                       ;  transfer.
  495 01FC B601501E                   rbit    t5stp,pwmdl   ; Start timer T5.
  496 0200 943A                       jmpl    tmrret        ; Return from interrupt.
  497
  498 0202 B6015210           t5wr9:  rbit    t5out,portpl  ; Remove EEPROM chip select, starting Write
  499                                                       ;  pulse within EEPROM.
  500 0206 874E1F0144AB               ld      t5,#TIMCON    ; Set up for delay of 20
  501                                                       ;  milliseconds (write pulse width).
  502 020C B601501E                   rbit    t5stp,pwmdl   ; Start timer T5.
  503 0210 942A                       jmpl    tmrret        ; Return from interrupt.
  504
  505 0212 B6015200           t5wr10: sbit    t5out,portpl  ; Set EEPROM chip select signal, ending Write
  506                                                       ;  pulse within EEPROM.
  507 0216 9701D6                     ld      sio,#x'01     ; Send Start bit for EWDS command (Disable
  508                                                       ;  Write/Erase).
  509 0219 835A0144AB                 ld      t5,#90        ; Set up for interrupt at end of MICROWIRE
  510                                                       ;  transfer.
  511 021E B601501E                   rbit    t5stp,pwmdl   ; Start timer T5.
  512 0222 59                         jmpl    tmrret        ; Return from interrupt.
  513
  514 0223 9700D6             t5wr11: ld      sio,#x'00     ; Send body of EWDS command.
```

TL/DD/9978–19

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)          EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                           PAGE  18
Timer Interrupt Handler

  515 0226 835A0144AB                 ld      t5,#90        ; Set up for interrupt at end of MICROWIRE
  516                                                       ;  transfer.
  517 022B B601501E                   rbit    t5stp,pwmdl   ; Start timer T5.
  518 022F 4C                         jmpl    tmrret        ; Return from interrupt.
  519
  520 0230 B6015210           t5wr12: rbit    t5out,portpl  ; Remove EEPROM chip select signal.
  521 0234 B6015010                   rbit    t5tie,pwmdl   ; Disable Timer T5 interrupts.
  522 0238 962E0F          R          sbit    nvravl,nvrs   ; Set EEPROM Available.
  523                         ;*** Here you'll want to set a semaphore bit saying that the WRITE
  524                         ;***  transfer is done.
  525 023B 40                         jmpl    tmrret
  526
  527 023C 3FC0               tmrret: pop     psw.w         ; Restore context.
  528 023E 3FC8                       pop     A
  529 0240 3E                         reti
  530
  531 0241                            .end    start
```

TL/DD/9978–20

```
ah      00C9 Abs Byte
al      00C8 Abs Byte
b2stp   0007 Abs Null
b8or16  0004 Abs Null
b8or9   0004 Abs Null
bfun    00F4 Abs Word
bfunh   00F5 Abs Byte
bfunl   00F4 Abs Byte
bh      00CD Abs Byte
bl      00CC Abs Byte
dirah   00F1 Abs Byte
dirb    00F2 Abs Word
dirbh   00F3 Abs Byte
dirbl   00F2 Abs Byte
divby   018E Abs Word
divbyh  018F Abs Byte
divbyl  018E Abs Byte
doeerr  0007 Abs Null
ei      0007 Abs Null
eiack   0002 Abs Null
eicon   015C Abs Word
eimode  0001 Abs Null
eipol   0000 Abs Null
enir    00D0 Abs Byte
enu     0120 Abs Byte
enui    0122 Abs Byte
enur    0128 Abs Byte
eri     0001 Abs Null
eti     0000 Abs Null
frmerr  0006 Abs Null
gie     0000 Abs Null
i0      0000 Abs Null
i2      0002 Abs Null
i3      0003 Abs Null
i4      0004 Abs Null
ibuf    00F0 Abs Byte
ircd    00D4 Abs Byte
irpd    00D2 Abs Byte
la0     0002 Abs Null
minit   006C Rel Null ROM16
```



```
nvradr  0000 Abs Null
nvravl  0007 Abs Null
nvrbuf  0020 Rel Word BASE
nvrbyt  0004 Abs Null
nvrcmd  002C Rel Byte BASE
nvrdta  ABCD Abs Null
nvrnum  002D Rel Byte BASE
nvrptr  0028 Rel Word BASE
nvrs    002E Rel Byte BASE
nvrwr   0006 Abs Null
nvrx    009A Rel Null ROM16
nvword  002A Rel Word BASE
obuf    00E0 Abs Byte
portah  00E1 Abs Byte
portb   00E2 Abs Word
portbh  00E3 Abs Byte
portbl  00E2 Abs Byte
portd   0104 Abs Byte
porti   00D8 Abs Byte
portp   0152 Abs Word
portph  0153 Abs Byte
portpl  0152 Abs Byte
psw     00C0 Abs Word
pwmdh   0151 Abs Byte
pwmdl   0150 Abs Byte
pwmode  0150 Abs Word
r1      0184 Abs Word
r2      0186 Abs Word
r3      018A Abs Word
r4      0142 Abs Word
r5      0146 Abs Word
r6      014A Abs Word
r7      014E Abs Word
rbfl    0001 Abs Null
rbit9   0003 Abs Null
rbuf    0124 Abs Byte
rdrdy   0001 Abs Null
rnvr    0079 Rel Null ROM16
runsys  0073 Rel Null ROM16
sio     00D6 Abs Byte
```

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)          EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                             PAGE  21
Timer Interrupt Handler

 sk      0006 Abs Null
 snvr1   0032 Rel Null ROM16
 snvr2   003E Rel Null ROM16
 so      0005 Abs Null
 sram    0007 Rel Null ROM16
 sraml1  000A Rel Null ROM16
 sraml2  0012 Rel Null ROM16
 stackb  0000 Rel Word BASE
 start   0000 Rel Null ROM16
 suwire  0015 Rel Null ROM16
 suwlp   002D Rel Null ROM16
 suwlp1  0039 Rel Null ROM16
 TIMCON  4E1F Abs Null
 t0ack   0003 Abs Null
 t0con   0192 Abs Byte
 t0pnd   0001 Abs Null
 t0tie   0000 Abs Null
 t1      0182 Abs Word
 t1ack   0007 Abs Null
 t1pnd   0005 Abs Null
 t1stp   0006 Abs Null
 t1tie   0004 Abs Null
 t2      0188 Abs Word
 t2ack   0003 Abs Null
 t2in    0003 Abs Null
 t2pnd   0001 Abs Null
 t2stp   0002 Abs Null
 t2tie   0000 Abs Null
 t3      018C Abs Word
 t3ack   0007 Abs Null
 t3pnd   0005 Abs Null
 t3stp   0006 Abs Null
 t3tie   0004 Abs Null
 t4      0140 Abs Word
 t4ack   0003 Abs Null
 t4out   0000 Abs Null
 t4pnd   0001 Abs Null
 t4stp   0002 Abs Null
 t4tfn   0003 Abs Null
 t4tie   0000 Abs Null
```

TL/DD/9978–23

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)          EEPROM                                  03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                                             PAGE  22
Timer Interrupt Handler

 t5      0144 Abs Word
 t5ack   0007 Abs Null
 t5int   00B8 Rel Null ROM16
 t5out   0004 Abs Null
 t5pnd   0005 Abs Null
 t5poll  00B2 Rel Null ROM16
 t5rd    00C5 Rel Null ROM16
 t5rd0   00D8 Rel Null ROM16
 t5rd1   00EB Rel Null ROM16
 t5rd2   00FA Rel Null ROM16
 t5rd3   0109 Rel Null ROM16
 t5rd4   011B Rel Null ROM16
 t5rddn  0132 Rel Null ROM16
 t5rdh   012D Rel Null ROM16
 t5rnxt  013C Rel Null ROM16
 t5stp   0006 Abs Null
 t5tfn   0007 Abs Null
 t5tie   0004 Abs Null
 t5wr    0148 Rel Null ROM16
 t5wr0   016A Rel Null ROM16
 t5wr1   017C Rel Null ROM16
 t5wr10  0212 Rel Null ROM16
 t5wr11  0223 Rel Null ROM16
 t5wr12  0230 Rel Null ROM16
 t5wr2   018A Rel Null ROM16
 t5wr3   01A1 Rel Null ROM16
 t5wr4   01B3 Rel Null ROM16
 t5wr5   01C3 Rel Null ROM16
 t5wr6   01D5 Rel Null ROM16
 t5wr7   01E6 Rel Null ROM16
 t5wr8   01F4 Rel Null ROM16
 t5wr9   0202 Rel Null ROM16
 t6      0148 Abs Word
 t6ack   0003 Abs Null
 t6out   0000 Abs Null
 t6pnd   0001 Abs Null
 t6stp   0002 Abs Null
 t6tfn   0003 Abs Null
 t6tie   0000 Abs Null
 t7      014C Abs Word
```

TL/DD/9978–24

```
NSC ASMHPC, Version E2 (Nov 02 15:51 1987)          EEPROM                    03-May-88 10:53
HPC-Based Driver for NMC9306/9345                                               PAGE  23
Timer Interrupt Handler

 t7ack   0007 Abs Null
 t7out   0004 Abs Null
 t7pnd   0005 Abs Null
 t7stp   0006 Abs Null
 t7tfn   0007 Abs Null
 t7tie   0004 Abs Null
 tbmt    0000 Abs Null
 tbuf    0126 Abs Byte
 tminit  0042 Rel Null ROM16
 tmmdh   0191 Abs Byte
 tmmdl   0190 Abs Byte
 tmmode  0190 Abs Word
 tmrint  00AE Rel Null ROM16
 tmrret  023C Rel Null ROM16
 tmrs    0005 Abs Null
 txd     0000 Abs Null
 uart    0006 Abs Null
 upic    00E6 Abs Byte
 upien   0003 Abs Null
 uwdone  0000 Abs Null
 uwmode  0001 Abs Null
 wakeup  0002 Abs Null
 wnvr    008C Rel Null ROM16
 wrrdy   0000 Abs Null
 xbit9   0005 Abs Null
 xh      00CF Abs Byte
 xl      00CE Abs Byte
 xrclk   0003 Abs Null
 xtclk   0002 Abs Null

 **** Errors:    0, Warnings:    0
```

TL/DD/9978–25

Lit. # 100552

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408