# Interfacing A Serial EEPROM to the National HPC16083

National Semiconductor Application Note 552 Brian Marley September 1988



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/PLUSTM 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.

# ABSTRACT

This application note describes how to interface the HPC16083 High-Performance microController to a MI-CROWIRE™ 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



Interfacing A Serial EEPROM to the National HPC16083

© 1995 National Semiconductor Corporation TL/DD/9978

RRD-B30M105/Printed in U. S. A

AN-55

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 MI-CROWIRE 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:

(ignored)

1

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" = Address bits,

most-significant bit first)

a pause of 16 to 25 milliseconds, with CS

low,

a WRITE command: 00000001 01AAAAAA DDDDDDDD DDDDDDD

("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 1 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.)

	IMC9306/9345	EEPROM	Ø3-May-88 10:53 PAGE 1
1	.titl	e EEPROM,'HPC-Based Driver for NMC9306/9345'	
3	; This code i ; or the 10	s written to drive either the 256-bit NMC93Ø6 (( 124-bit NMC9345 (COP495) MICROWIRE(tm) EEPROM.	COP494)
5 6 7 8 9	; NOTE: Timi ; cr ; wi ; ac	ng values assume that the HPC is running at 16M ystal frequency. For correct programming pulse dths, one should not deviate far from this with djusting the timing constant below.	H <b>z</b> Dut
10 11 4E1F 12 13 14	TIMCON =	19999 ; 20000 counts at 1 usec = 20 msec. ; Timing constant for ERASE and WRI ; pulse widths.	TE
			TL/DD/997
NSC ASMHPC, Version E HPC-Based Driver for I Declarations: Regist	2 (Nov Ø2 15:51 1987) NMC93Ø6/9345 er Addresses	EEPROM	<b>Ø3-May-88</b> 10:53 PAGE 2
15	.for	m 'Declarations: Register Addresses'	
17 00C0 18 00C8 19 00C9 20 00CC 21 00CC 22 00CC	psw = at = ah = bl = bh = xt =	x'CØ:w ; PSW register x'C3:b ; Low byte of Accumulator. x'C9:b ; High byte of Accumulator. x'CC:b ; Low byte of Register B. x'CC:b ; Low byte of Register B. x'CE:b ; Low byte of Register X.	
23 00CF 24 25 00D0 26 00D2 27 00D4 28 00D6	xh = enir = irpd = ircd = sio =	x'CF:b ; High Byte of Register X. x'DØ:b x'D2:b x'D4:b x'D6:b	
29 0008 30 0000 31 0001 32 0002 33 0002 34 0003	porti = obuf = portah = portb = portbl = portbh =	x'D8:b x'E0:b ; (Low byte of PORTA.) x'E1:b ; High byte of PORTA. x'E2:w x'E2:b ; Low byte of PORTB. x'E3:b ; High byte of PORTB.	
35 00E6 36 00F0 37 00F1 38 00F2 39 00F2 40 00F3	upic = ibuf = dirah = dirb = dirbl = dirbl =	x'E6:b x'FØ:b ; (Low byte of DIRA.) x'F1:b ; High byte of DIRA. x'F2:w x'F2:b ; Low byte of DIRB.	
41 00F4 42 00F4 43 00F5 44	bfun = bfunl = bfunh =	x1F4:w x1F4:b ; Low byte of BFUN. x1F5:b ; High byte of BFUN.	
45 0104 46 0120 47 0122 48 0124 49 0126 50 0128	portd = enu = enui = rbuf = tbuf = enur =	x'9194:b x'9129:b x'9122:b x'9124:b x'9126:b x'9126:b	
51 52 Ø14Ø 53 Ø142	t4 = r4 =	x*0140:w x*0142:w	
54 Ø144	to =	X ' 19 1 4 4 : W	TL/DD/997

Declarations: Register 55 Ø146 56 Ø148 57 Ø14A 58 Ø14C 59 Ø14E 60 Ø150 61 Ø150 62 Ø151 63 Ø152 64 Ø152 65 Ø153 66 Ø15C 67	Addresses r5 = x' r6 = x' r7 = x' pumode = x' pumod = x' pumod = x' portp = x' portpl = x' eicon = x'	0146:w 0148:w 0148:w 0146:w 0150:b ; Low byte of PWMODE. 0150:b ; High byte of PWMODE. 0152:w 0152:b ; Low byte of PORTP. 0153:b ; High byte of PORTP. 0155:w ; High byte of PORTP.	
68 Ø182 69 Ø184 70 Ø186 71 Ø186 73 Ø186 73 Ø186 74 Ø188 75 Ø186 75 Ø186 76 Ø186 76 Ø187 78 Ø190 79 Ø191 80 Ø192 81	<pre>t1 = x' r1 = x' r2 = x' t2 = x' t3 = x' divby = x' divby = x' divbyl = x' divbyh = x' timmode = x' timmodh = x' timmodh = x' timodh = x'</pre>	0182:w 0184:w 0186:w 0188:w 0188:w 0186:w 0186:b ; Low byte of DIVBY. 0186:b ; High byte of DIVBY. 0199:b ; Low byte of TMMODE. 0199:b ; High byte of TMMODE. 0192:b	
			TL/DD/99
NSC ASMHPC, Version E2 HPC-Based Driver for NMM Declarations: Bit Posit	(Nov Ø2 15:51 1987) C93Ø6/9345 tions	EEPROM	Ø3-May-88 10:53 PAGE 4
83 84	.form 'D	eclarations: Bit Positions'	
85 86	Name Positi	on Register(s)	
87			
87 88 0000 90 0002 91 0003 92 0004 93 0005 94 0006 95 0007 96	gie = Ø iØ = Ø i2 = 2 i3 = 3 i4 = 4 tmrs = 5 uart = 6 ei = 7	; enir ; porti only ; enir, irpd, ircd ; enir, irpd, ircd ; enir, irpd, ircd ; enir, irpd ; enir, irpd ; enir, irpd	
87 88 ØØØØ 99 ØØØ2 91 ØØ02 92 ØØØ4 93 ØØØ5 94 ØØ06 95 ØØØ7 96 97 ØØØ1 98 ØØØØ 99	gie = Ø iØ = Ø i2 = 2 i3 = 3 i4 = 4 tmrs = 5 uart = 6 ei = 7 uwmode = 1 uwdone = Ø	; enir ; porti only ; enir, irpd, ircd ; enir, irpd, ircd ; enir, irpd ; enir, irpd ; enir, irpd ; enir, irpd ; ircd ; irpd	
87 88 0000 90 0002 91 0003 92 0004 93 0005 94 0006 95 0007 96 97 0001 98 0000 101 0001 102 0004 105 0005 104 0005 104 0005 105 0003 105 0005 106 0006 107 0007 108 0000 107 0007 108 0000 109 0001 110 0002 129 0001	gie = 0 i0 = 0 i2 = 2 i3 = 3 i4 = 4 tmrs = 5 uart = 6 ei = 7 uwmode = 1 uwdone = 0 tbmt = 0 rbfl = 1 b8or9 = 4 xbit9 = 2 rbit9 = 3 frmerr = 6 doeerr = 7 eti = 1 xtclk = 3 b2stp = 7	; enir ; porti only ; enir, irpd, ircd ; enir, irpd, ircd ; enir, irpd ; enir, irpd ; enir, irpd ; enir, irpd ; ircd ; ircd ; enu ;	
87 88 \$P\$P\$P 99 \$P\$P\$P 90 \$P\$P\$P 91 \$P\$P\$P 93 \$P\$P\$P 93 \$P\$P\$P 94 \$P\$P\$P 95 \$P\$P\$P 97 \$P\$P\$P 98 \$P\$P\$P 99 99 100 \$P\$P\$P 101 \$P\$P\$P 102 \$P\$P\$P 104 \$P\$P\$P 105 \$P\$P\$P 105 \$P\$P\$P 105 \$P\$P\$P 106 \$P\$P\$P 106 \$P\$P\$P 107 \$P\$P\$P 108 \$P\$P\$P 109 \$P\$P\$P 101 \$P\$P\$P\$P 101 \$P\$P\$P\$P 101 \$P\$P\$P\$P 101 \$P\$P\$P\$P\$P 101 \$P\$P\$P\$P\$P\$P 100 \$P\$P\$P\$P 100 \$P\$P\$P\$P\$P 100 \$P	gie       =       Ø         i2       =       2         i3       =       3         i4       =       4         tmrs       =       5         uart       =       6         ei       =       7         uwmode       =       1         uwdone       =       Ø         rbfl       =       1         b8or9       =       4         xbit9       =       5         wakeup       =       2         rbit9       =       3         frmerr       =       6         doeerr       =       7         wrclk       =       2         rclk       =       3         b2stp       =       7         wrdy       =       Ø         rdrdy       =       1         iaØ       =       2         upien       =       3         b8or16       =       4	; enir ; porti only ; enir, irpd, ircd ; enir, irpd, ircd ; enir, irpd ; enir, irpd ; enir, irpd ; enir, irpd ; ircd ; ircd ; ircd ; enu ;	
87 88 \$P\$P\$P 99 \$P\$P\$P 91 \$P\$P\$P 93 \$P\$P\$P 93 \$P\$P\$P 93 \$P\$P\$P 94 \$P\$P\$P 95 \$P\$P\$ 96 97 98 \$P\$P\$P 98 \$P\$P\$ 99 99 99 99 99 99 99 99 99 99 99 99 99	gie       =       Ø         i2       =       2         i3       =       3         i4       =       4         tmrs       =       5         uart       =       6         ei       =       7         uwmode       =       1         uwdone       =       Ø         rbfl       =       1         b8or9       =       4         xbit9       =       5         wakeup       =       2         rbit9       =       3         frmerr       =       6         doerr       =       7         wrclk       =       2         xrclk       =       3         b2stp       =       7         wrdy       =       Ø         vrdy       =       Ø         upien       =       3         b8or16       =       4         tØrtie       =       Ø         tØrdy       =       1         tØrde       =       1	; enir ; porti only ; enir, irpd, ircd ; enir, irpd, ircd ; enir, irpd ; enir, irpd ; enir, irpd ; enir, irpd ; ircd ; ircd ; ircd ; enu ;	

Declarations: Bit Po	sitions			
123 0004	t1tie =	4	; tmmdl	
124 0005	t1pnd = t1stn ≃	5	; tmmdl • tmmdl	
126 0007	tlack =	7	; tmmdl	
127 0000 128 0001	t2tie =	Ø 1	; tmmdh	
129 0002	t2stp =	ź	; tmmdh	
130 0003	t2ack =	3	; tmmdh	
132 0005	t3pnd =	5	; tmmch	
133 0006	t3stp =	6	; tmmdh	
134 10/07	t3ack =	1	; tmmdh	
136 0000	t4tie =	ø	; pwmdl	
137 0001 138 0002	t4pnd = t4stn =	1	; pwmdl	
139 0003	t4ack =	3	; pwmdt	
140 0004	t5tie =	4	; pwmdl	
142 0006	t5stp =	6	; pwmdl	
143 0007	t5ack =	7	; pwmdl	
144 0000	tótie ≂ tópnd =	1	; pwmdh : pwmdh	
146 0002	tóstp =	ź	; pwmdh	
147 9993 148 9994	tóack = t7tie =	3	; pwmdh : pwmdh	
149 0005	t7pnd =	5	; pwmdh	
150 0006	t7stp =	6	; pwmdh	
152	track =	'	; pwildn	
153 0000	t4out =	ę	; portpl	
154 9995	t4tfn = t5out =	3	; portpl ; portpl	
156 0007	t5tfn =	7	; portpl	
157 0000 158 0003	t6out =	Ø	; portph	
159 0004	t7out =	4	; portph	
160 0007 161	t7tfn =	7	; portph	
162 9999	eipol =	0	: eicon	
NSC ASMHPC. Version F	2 (Nov 02 15:51 1987)	2	FEDDON	TL/DD/9
NSC ASMHPC, Version E2 HPC-Based Driver for M Declarations: Bit Pos	2 (Nov Ø2 15:51 1987) NMC93Ø6/9345 sitions	2	EEPROM	TL/DD/9 <b>93-May-88 19:53</b> PAGE 6
NSC ASMHPC, Version E2 HPC-Based Driver for N Declarations: Bit Pos 163 ØØØ1 164 ØØØ2	2 (Nov Ø2 15:51 1987) NMC93Ø6/9345 sitions eimode = eiack =	1 2	; eicon	TL/DD/9 <b>93-May-88 19:53</b> PAGE 6
NSC ASMHPC, Version E2 HPC-Based Driver for N Declarations: Bit Pos 163 0001 164 0002	2 (Nov Ø2 15:51 1987) MMC93Ø6/9345 sitions eimode = eiack =	1 2	EEPROM ; eicon ; eicon	TL/DD/9 <b>93-May-88 19:53</b> PAGE 6
NSC ASMHPC, Version E2 HPC-Based Driver for M Declarations: Bit Pos 163 ØØØ1 164 ØØØ2 165 166 ØØØØ 167 ØØØ3	2 (Nov Ø2 15:51 1987) WNC93Ø6/9345 sitions eimode = eiack = txd = t2in =	12	; eicon ; eicon ; eicon ; portbl, dirbl, bfunl	TL/DD/9 <b>03-May-88 10:53</b> PAGE 6
NSC ASMHPC, Version E PPC-Based Driver for M Peclarations: Bit Pos 163 0001 164 0002 165 166 0000 167 0003 168 0005	2 (Nov Ø2 15:51 1987) NMC93Ø6/9345 sitions eimode = eiack = txd = t2in = so =	1 2 9 3 5	; eicon ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>03-May-88 10:53</b> PAGE 6
NSC ASMHPC, Version E; IPC-Based Driver for N Declarations: Bit Pos 163 0001 164 0002 165 166 0000 167 0003 168 0005 169 0005 169 0005 169 0005 169 0005 169 0005 169 0005	2 (Nov 02 15:51 1987) WMC9306/9345 sitions eimode = eiack = txd = t2in = sk =	1 2 3 5 6	; eicon ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>03-May-88 10:53</b> PAGE 6
NSC ASMHPC, Version E2 HPC-Based Driver for M Declarations: Bit Pos 163 0001 164 0002 165 0000 167 0003 168 0005 169 0005 169 0005 170	2 (Nov Ø2 15:51 1987) NMC93Ø6/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 3 5 6	; eicon ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 Ø3-May-88 10:53 PAGE 6 TL/DD/9
NSC ASMHPC, Version E2 HPC-Based Driver for N Declarations: Bit Pos 163 0001 164 0002 165 165 166 0000 167 0003 168 0005 169 0005 169 0005 170 171	2 (Nov Ø2 15:51 1987) NMC93Ø6/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 3 5 6	; eicon ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>Ø3-May-88 19:53</b> PAGE 6 TL/DD/9
NSC ASMHPC, Version E; HPC-Based Driver for N Declarations: Bit Pos 163 0001 164 0002 165 166 0000 167 0003 168 0005 169 0006 170 171	2 (Nov 02 15:51 1987) NNC9306/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 3 5 6	; eicon ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>93-May-88 19:53</b> PAGE 6 TL/DD/9
NSC ASMHPC, Version E2 HPC-Based Driver for M Declarations: Bit Pos 163 0001 164 0002 165 166 0000 167 0003 168 0005 169 0005 169 0006 170 171	2 (Nov Ø2 15:51 1987) WNC93Ø6/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 9 3 5 6	EEPROM ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>93-May-88 19:53</b> PAGE 6 TL/DD/9
NSC ASMHPC, Version E2 HPC-Based Driver for N Declarations: Bit Pos 163 0001 164 0002 165 165 166 0000 167 0005 169 0005 169 0005 170 171	2 (Nov Ø2 15:51 1987) NMC93Ø6/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 8 3 5 6	EEPROM ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>Ø3-May-88 10:53</b> PAGE 6 TL/DD/9
NSC ASMMPC, Version E; HPC-Based Driver for M Declarations: Bit Pos 163 0001 164 0002 165 166 0000 167 0003 168 0005 169 0006 170 171	2 (Nov 02 15:51 1987) WMC9306/9345 sitions eimode = eiack = txd = t2in = sk =	1 2 8 3 5 6	EEPROM ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>Ø3-May-88 19:53</b> PAGE 6 TL/DD/9
NSC ASMHPC, Version E2 HPC-Based Driver for M Declarations: Bit Pos 163 0001 164 0002 165 0003 167 0003 168 0005 169 0005 170 171	2 (Nov Ø2 15:51 1987) NMC93Ø6/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 3 5 6	EEPROM ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 Ø3-May-88 19:53 PAGE 6 TL/DD/9
NSC ASMHPC, Version E2 HPC-Based Driver for N Declarations: Bit Pos 163 0001 164 0002 165 166 0000 167 0003 168 0005 169 0005 169 0005 170 171	2 (Nov Ø2 15:51 1987) NMC93Ø6/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 8 3 5 6	EEPROM ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>Ø3-May-88 19:53</b> PAGE 6 TL/DD/9
NSC ASMMPC, Version E; MPC-Based Driver for M Declarations: Bit Pos 163 0001 164 0002 165 166 0003 167 0003 168 0005 169 0006 170 171	2 (Nov 02 15:51 1987) NMC9306/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 8 3 5 6	EEPROM ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>93-May-88 19:53</b> PAGE 6 TL/DD/9
NSC ASMHPC, Version E2 HPC-Based Driver for M Declarations: Bit Pos 163 0001 164 0002 165 0000 167 0003 168 0005 169 0006 170 171	2 (Nov Ø2 15:51 1987) MMC93Ø6/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 8 3 5 6	EEPROM ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	ТL/DD/9 Ø3-May-88 19:53 PAGE 6 TL/DD/9
NSC ASMHPC, Version E; HPC-Based Driver for N Declarations: Bit Pos 163 0001 164 0002 165 166 0000 167 0003 169 0005 169 0006 170 171	2 (Nov 02 15:51 1987) NMC9306/9345 sitions eimode = eiack = txd = t2in = so = sk =	1 2 8 3 5 6	EEPROM ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 <b>93-May-88 19:53</b> PAGE 6 TL/DD/9
NSC ASMMPC, Version E2 PPC-Based Driver for M Declarations: Bit Pos 163 0001 164 0002 165 166 0003 167 0003 168 0005 169 0005 170 171	2 (Nov 02 15:51 1987) WMC9306/9345 sitions eimode = eiack = txd = t2in = sk = sk =	1 2 3 5 6	EEPROM ; eicon ; eicon ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl ; portbl, dirbl, bfunl	TL/DD/9 Ø3-May-88 19:53 PAGE 6 TL/DD/9

HPC-Bas Space D	sed Dr Declar	iver for NMC9306/93 ations	45	07)	EEPROM		PAGE 7
172 173	9999			.form .sect	'Space Declarat DSECT,BASE,REL	ions'	
174 175			;WORD-A	LIGNED V	ARIABLES		
176 177	0000		stackb:	.dsw	16 ; Space	for 16 words.	
178	0020		nvrbuf:	.dsw	4 ; EEPRO	M String Buffer. FERROM Data buffer	
180	002A		nvword:	.dsw 1	; Scratch locat	ion for gathering EEPROM data as wo	rds.
181 182			;BYTE-A	LIGNED V	ARIABLES		
183 184	ØØ2C		nvrcmd:	.dsb 1	; Current EEPRO	M command.	
185 186	002D 002F		nvrnum:	.dsb 1 .dsb 1	; Byte count fo : EEPROM status	er current EEPROM Read command. byte: phase number for sequencing	MICROWIRE
187	ppee				; transfers.	-,,	
189			;BIT DE	FINITION	s		
19Ø 191				; NVRS	byte: Status of	EEPROM MICROWIRE transfers.	
192 193				÷	Contair in low	ns phase (step number) of current EE order 4 bits. Top two bits are as	PROM command follows:
194	0007 0004		nvravl=	7 6	; When set, ind	licates that no EEPROM command is in	progress. FFPROM Write
196	μμφu			5	, y nouns an Et	and head to the progressy in media	
197							TL/DD/9
ISC ASM IPC-Bas	HPC, sed Dr	Version E2 (Nov Ø2 iver for NMC93Ø6/93	15:51 19 45	87)	EEPROM		Ø3-May-88 10:53 PAGE 8
Code Se	ection						
198 199	0000			.form .sect	'Code Section' CSECT,ROM16,REL	; Code space.	
200	0000	870008C0	start.	Id	DSW #x108	· Set one WAIT state	
202	0004	970000	300101	ld	enir,#x'00	; Disable interrupts	
203						, narviduatty.	
205 206	0007		sram:		; Clear ; Baser	all RAM locations. Dage bank:	
207 208	0007 8	80ØØBE 80	sraml1.	ld clr	BK,#X'0000,#X'0	ØBE ; Establish loop base and	limit.
209	000B	E1		xs	A, [B+] .w		
211	DUDC (	~~		14	. n. 1		
212	0000 I	A701C001FE		ld	; Non-t BK,#x'91C0,#x'9	asepage Dank: 1FE ; Establish loop base and	limit.
214 215	ØØ12   ØØ13	00 E 1	sraml2:	clr xs	A A,[B+].w		
216	0014	62		jp	sraml2		
218	0015		suwire:			; MICROWIRE setup.	
219						; deselected on reset, since	
221 222						; Port P is cleared.)	
223	0015 0018	96F4ØD 96F20D		sbit sbit	so,bfunl so.dirb!	; Enable SO output.	
225	001B	96E21E		rbit	sk,portbl	; Set up SK output.	
226	ØØ21	96F4ØE		sbit	sk, bfunl		
228 229	ØØ24 ØØ27	960409 872225018EAB		sbit ld	uwmode,ircd divby,#x'2225	; Set Master Mode. ; Set MICROWIRE frequency.	
230 231	002D	960210	suwin:	ifbit	uwdone.inpd	: Wait until MICROWIRE	
232	0030 ·	41		jp	snvr1	; interface ready (uWDONE bit set)	
233	00	U~	<b>-</b>	14	sawip		
235 236 237	0032 0032 0036	8601520C 970006	snvr1:	sbit ld	t5out,portpl sio,#9	; cancel any EERKUM Write in progr ; Set EEPROM Chip Select active. ; Send a byte of zeroes.	255:
							TL/DD/997

258       859       900218       sudp1: fibit       undersign       : Unit unit MCROME         258       850       10       product       interface       interface         258       850       10       product       interface       interface         258       850       10       station       interface       interface         258       850       10       thintit       it foot, product       it foot, product       it foot, product         258       860       8555816648       id       intoole, #foot       it foot, product       it foot, pr	NSC ASM HPC-Bas Code Se	HHPC, sed Di ection	version E2 (Nov river for NMC930 n	/ 02 15:51 19 06/9345	0()	EEPROM	3M-CQ	PAGE 9
21       B32E B051521C       env2: fbit       fSuf_portpl       : Recorpt # 10         23       B42C 85568175200       tminit: [d       tRoot # 40       : Stop times 11, 12, 13.         23       B42S 8558175200       tminit: [d       tRoot # 40       : Stop times 11, 12, 13.         23       B42S 87CCC89198A8       [d       tmode, # 4CC20       : Tel21 internet       : to CU/128.         23       B42S 87CCC89198A8       [d       pamode, # 4CC20       : Tel23 internet 1-17.       : Tel23 internet 1-17.         23       B43S 44       moop       pamode, # 4CC20       : Tel23 internet 1-17.       : Tel23 internet 1-17.         23       B43S 44       moop       pamode, # 4CC20       : Tel23 internet 1-17.       : Tel23 internet 1-17.         23       B43S 47CCC815A8       [d       pamode, # 4CC20       : Tel23 internet 1-17.       : Tel23 internet 1-17.         23       B44S 87CCC815A8       [d       rs, # 4C44       : No modulus for EEPROM timer.       : Tel23 internet 1-17.         25       B44S 87CCC815A8       [d       rs, # 47.88       : No modulus for EEPROM timer.       : Tu00/00         25       moop       . form       'maint:       : No modulus for EEPROM timer.       : Tu00/00         25       # 46       . form	238 239 249	0039 003C	960210 41 64	suwlp1:	ifbit jp ip	uwdone,irpd snvr2 suwlp1	; Wait until MICROWIRE ; interface ready (uWDONE ; bit set).	
223       BAC2 2336919228       tminitid       timode #X464       Stop timers 11, 12, 13.         224       BAC2 2366910283       td       timode #X464       Stop timers 11, 12, 13.         224       BAC2 2359192848       td       timode #X464       Stop timers 11, 12, 13.         225       BAC2 2359192848       td       timode #X464       Stop timers 11, 12, 13.         225       BAC2 23591928       td       timode #X464       Stop timers 11, 12, 13.         226       BAC2 23581928       td       timode #X464       Stop timers 11, 12, 13.         226       BAC2 2358       td       pamode #X464       Stop timers 11, 12, 13.         226       BAC2 2358       td       pamode #X464       Stop timers 11, 12, 13.         226       BAC2 2358       td       pamode #X464       Stop timers 11, 12, 13.         226       BAC2 2358       td       pamode #X464       Stop timers 11, 12, 13.         226       BAC2 2358       td       rstop timers 11, 12, 13.       tmode #X464         237       BAC2 2358       td       rstop timers 11, 12, 13.       tmode #X4764         238       BAC6 277672       R       intit       td       rstop timers 11, 12, 13.         238       BAC5 738262	241	003E	B691521C	snvr2:	rbit	t5out,portpl	; Remove EEPROM Chip Select.	
<pre>22</pre>	243 244 245	0042 0047 004D	830801928B 8744400190AB 8355018EAB	tminit:	ld ld ld	tØcon,#x'Ø8 tmmode,#x'444Ø divby,#x'ØØ55	; Stop timers T1, T2, T3. ; MICROWIRE frequency set	
251       BSD B7744449159AB       id       penode,#x14444       : Stop timers 1/-17.         252       BSD F 40       nop       it it it rending bits to       it i	246 247 248	ØØ52	87CCC80190AB		ld	tmmode,#x1CCC8	; to CK1/128. ; Clear and disable timer ; TØ-T3 interrupts.	
252     8957 49     Id     panode,#xtCCCC     file.ard drough derive treating them.       254     995     576     panode,#xtCCCC     file.ard drough derive treating them.       255     956     577 FFF9166A8     id     r5,#xtFFF     ; No modulus for EEPROM timer.       255     576     576     576     577 FFF9166A8     id     r5,#xtFFF     ; No modulus for EEPROM timer.       256     577 FFF9166A8     id     r5,#xtFFF     ; No modulus for EEPROM timer.     TL/DOVER       257     700 FFF69166A8     id     r5,#xtFFF     ; No modulus for EEPROM timer.     TL/DOVER       257     700 FFF69166A8     id     r5,#xtFFF     ; No modulus for EEPROM timer.     TL/DOVER       258     700 FFF69166A8     id     r5,#xtFFF     ; Set EEPROM existing the tree.     FARE 19       259     700 FF767     1607 FF767     1607 FF767     1607 FF767     1607 FF767     1607 FF767       259     973     runsys:     ; Enable timer interrupts, and go to main.     160 FF767     1607 FF767       256     9973     runsys:     ; Enable timer interrupts, doube tree     1700 FF77       270     9975 960980     sbit     gie,enir     ; Enable interrupt system.       271     9975 960983     sbit     gie,enir     ; Enable inte	249 250 251	ØØ58 ØØ5E	8744440150AB 40		ld nop	pwrnode,#x'4444	; Stop timers T4-T7. ; Wait for Pending bits to	
253 233       Ld       r5,#x'FFFF       ; No modulus for EEPROM timer.         TUDD/981         INTE-Based Driver for MKC3960/9358         Start S	252 253 254 255	005F 0060	40 87CCCC0150AB		nop ld	pwmode,#x'CCCC	; trickle through before clearing them. ; Clear and disable ; interrupts from all ; PWM timers.	
<pre>XEXASMPC, Version E2 (Nov #2 15:51 1987) EFROM B: Artor-88 19:53 Profestation  %</pre>	256 257 258	ØØ66	87FFFFØ146AB		۱d	r5,#x'FFFF	; No modulus for EEPROM timer.	
<pre>MSC ASMMPC, Version E2 (Nov 92 15:11 1987) EEPROM #3-Hay-88 18:33 PAGE 19 19 19 19 19 19 19 19 19 19 19 19 19 1</pre>								TL/DD/997
<pre>259 .form 'Hain Program Initialization' 264 006C TRADE TO THE TABLE TO THE TAB</pre>	NSC ASM HPC-Bas Main Pr	MHPC, sed Di rogram	Version E2 (Nov river for NMC93) n Initialization	v Ø2 15:51 19 Ø6/9345 n	87)	EEPROM	Ø3-M	ay-88 10:53 PAGE 10
<pre>page page page page page page page page</pre>	259				.form	'Main Program 1	Initialization'	
<pre>223 for the start of Duffer. 224 gP73 runsys: ; Enable timer interrupts, and go to main. 225 gP73 960 gP0 sbit tmrs,enir ; Enable timer interrupts. (Done here 226 gP76 960 gP8 sbit gie,enir ; Enable interrupt system. 277 279 gP76 960 gP8 sbit gie,enir ; Enable interrupt system. 271 272 272 272 272 272 272 272 272 272</pre>	261 262 263	006C 006C 006F	97802E 87002028	minit: R R	ld Id	nvrs,#x180 nvrstr #ovrbuf	; Set EEPROM available.	
<pre>226 / PATS 960800 sbit tmrs.enir ; Enable timer interrupts. (Women ere</pre>	264 265	0073				• Enabl	e timer interrunts and so to main	•
<pre>200 ; INITIALIZE command first.) 271 272 272 272 272 272 272 272 272 272</pre>	266 267 268	ØØ73	96DØØD	. 61375.	sbit	tmrs,enir	; Enable timer interrupts. (Done here ; to allow engine commands without an	
22 TLODAS	269 270 271	ØØ76	960008		sbit	gie,enir	; INITIALIZE command first.) ; Enable interrupt system.	
	212							TI /DD/997
								11/00/99/

NSC ASMHPC, Version E2 (Nov Ø2 15:51 1987) HPC-Based Driver for NMC93Ø6/9345 Main Program Fragments EEPROM Ø3-May-88 10:53 PAGE 11 .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. ;;;; Ø ; EEPROM address: change to suit your application.
x'ABCD ; Written data: change to suit.
4 ; Number of bytes to read (1-8): change to suit. nvradr = nvrdta nvrbvt = ; Read Fragment: reads up to 4 words (8 bytes) from EEPROM. ; Get NVR starting address. ; Truncate to legal limit. ; Create NVR READ command. ; Place it in memory. ; Get number of bytes requested. ; Save byte count in memory. ; Set up NVR access status flags: ; Read transfer in progress, first phase. ; Reset buffer pointer to beginning. ; Go start up transfer. A,#nvradr rnvr: ۱d and shi A,#x'3F R st Id st Id A nyrcmd A,#nvrbyt A,nvrnum R R nvrs,#Ø R ١d nvrptr,#nvrbuf jmpl nvrx ; Write Fragment: writes one word to EEPROM. nvword,#nvrdta ; Get data word. R whyr: Id ; Get data word. ; Get EFROM address. ; Mask it for proper range. ; Store it in Command byte in memory. ; (Opcode = ββ at this point.) ; Set up NVR access status flags: ; Write transfer in progress, first phase. ; Go start up transfer. A,#nvradr A,#x'3F id and R A.nvrcmd st ١d nvrs,#x'4Ø R jmpl nvrx ; Common routine, performed by both READ and WRITE. ; Start interrupts from Timer T5 to schedule ; accesses to EEPROM. nvrx: TL/DD/9978-13 NSC ASMHPC, Version E2 (Nov Ø2 15:51 1987) HPC-Based Driver for NMC93Ø6/9345 Main Program Fragments Ø3-May-88 10:53 PAGE 12 EEPROM ; Interrupts are not repetitive; give R5 a ; high value. ; Set Timer T5 to interrupt (almost) ; immediately when started. ; Enable interrupt from Timer T5. ; Start Timer T5. 313 009A 87FFFF0146AB 314 ١d r5,#x'FFFF 315 00A0 83000144AB ١d t5,#Ø 316 317 ØØA5 86Ø15ØØC 318 ØØA9 86Ø15Ø1E t5tie,pwmdl t5stp,pwmdl sbit rbit 319 320 321 \*\*\* 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 "\*\*\*". ;;;;; 322 323 324 325 00AD 60 ; Stops HPC, except for interrupt service. ip 326 327 328 END OF MAIN PROGRAM FRAGMENTS. ; TL/DD/9978-14

C-Bas	nter						
329					.form	'Timer Interrup	ht Handler'
330 331 332 333				;	The Tir inte oper	mer 15 interrupt errupt sequences ration in progres	service routine does all the work. Each the next step of the READ or WRITE s.
335	FFF4	AEØØ	R		.ipt	5,tmrint	; Declare entry point for Timer Interrupt.
336 337 338	00AE 00B0	AFC8 AFCØ		tmrint:	push push	A psw.w	; Save context.
340 341 342	ØØ82 ØØ86	B6015015 41		t5poll:	ifbit jmpl	t5pnd,pwmdl t5int	; Poll for Timer T5 interrupt (EEPROM Timing ; Interrupt).
343	ØØB7	60			jp		; Otherwise, error. Stop HPC.
344 345 346	ØØ88 ØØ8C	B601500E B601500F		t5int:	sbit sbit	t5stp,pwmdl t5ack,pwmdl	; Stop Timer T5. ; Clear interrupt request. (Doing this ; immediately is acceptable bare )
348	ØØCØ	962E16	R		ifbit	nvrwr,nvrs	; Check whether Read or Write operation is
349 350 351	ØØC3	9483		4 <b>5</b>	jmpl	t5wr	; is in progress. ; If Write, go perform ; Enable/Erase/Write/Disable operation.
352 353	00C5 00C5	882E	R	tora:	ld	A, nvrs	; Else, program is reading from EEPROM. ; Get phase info.
354 355 356	00C7 00C9 00CB	892E 99ØF E7	R		inc and shl	nvrs A,#x'ØF A	; Increment memory value for next T5 interrupt. ; Extract phase number. ; Jump based on this number.
357 358 359	00CD 00CD 00CD 00CE 00D0 00D0	49 EC 9A00 1B00 2800			.odd jidw .ptw	t5rdØ,t5rd1,t5r	d2,t5rd3,t5rd4
7/0	0004 0006	3500 4500					
2014							
361 362 363	0008 000C	B601520C 9703D6		t5rd <b>0</b> ∶	sbit ld	t5out,portpl sio,#x'Ø3	; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0,
361 362 363	0008 000C	8601520C 9703D6		t5rd <b>Ø:</b>	sbit ld	t5out,portpl sio,#x'Ø3	; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997
361 362 363 363 SC ASM C-Bas	99D8 99DC IHPC, sed D	B601520C 9703D6 Version E2 (N river for MMC9 rupt Handler	lov Ø2 23Ø6/93	t5rdØ: 15:51 19 45	sbit ld 87)	t5out,portpl sio,#x'Ø3 EEPROM	; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997 03-May-88 10:53 PAGE 14
361 362 363 SC ASM C-Bas imer 1 364 365 366 367 368	ØØD8 ØØDC IHPC, sed D inter	B601520C 9703D6 Version E2 (N river for NMC9 rupt Handler	lov Ø2 23Ø6/93	t5rdØ: 15:51 19 45	sbit ld 187)	t5out,portpl sio,#x*β3 ΕΕΡRΟΜ	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>
361 362 363 SC ASI C-Bas imer 1 364 365 366 367 368 369 370 371	ØØD8 ØØDC inter ØØDF ØØE4 ØØE8	B601520C 9703D6 Version E2 (N river for NMC9 rupt Handler 835A0144AB B601501E B40151	40v Ø2 93Ø6/93	t5rdØ: 15:51 19 45	sbit ld 87) ld rbit jmpl	t5out,portpl sio,#x*Ø3 EEPROM t5,#90 t5stp.pumdl tmrret	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>
361 362 363 363 363 363 364 365 364 365 366 366 367 368 369 371 372 373	ØØD8 ØØDC IHPC, inter ØØDF ØØE4 ØØE8 ØØE8	B601520C 9703D6 Version E2 (N river for NMC9 rupt Handler 835A0144AB B601501E B40151 8C2CD6	Nov Ø2 93Ø6/93 R	t5rdØ: 15:51 19 45 t5rd1:	sbit ld 187) 87) ld jmpl ld	t5out,portpl sio,#x'Ø3 EEPROM t5,#90 t5stp,pwmdl tmrret sio,nvrcmd	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>
369 361 362 363 362 363 362 363 364 365 364 365 366 367 368 371 373 374 375 376 377	ØØD8 ØØDC IHPC, ied D inter ØØDF ØØE8 ØØE8 ØØE8 ØØE8 ØØF3 ØØF7	B601520C 9703D6 Version E2 (N river for NMC9 rupt Handler 835A0144AB B601501E B40151 8C2CD6 835A0144AB B601501E 840142	Nov Ø2 93Ø6/93 R	t5rdØ: 15:51 19 45 t5rd1:	sbit ld 187) 187) Id ld ld ld ld ld ld	t5out,portpl sio,#x*Ø3 EEPROM t5stp,pwmdl tmrret sio,nvrcmd t5,#90 t5stp,pwmdl tmrret	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>
367 361 362 363 362 363 364 366 366 366 366 366 370 370 371 373 374 375 375 376 377 378 380 380 382	0008 0000 0000 0000 0000 0000 0000 000	B601520C 9703D6 Version E2 (N river for NMC9 rupt Handler 835A0144AB B601501E B40151 8C2CD6 835A0144AB B601501E B40142 970006 835A0144AB B601501E 840133	Nov Ø2 93Ø6/93 R	t5rd9: 15:51 19 45 t5rd1: t5rd2:	sbit ld (87) (87) ka rbit jmpl ld ld ld ld rbit jmpl ld ld ka rbit jmpl	t5out, portpl sio, #x*93 EEPROM t5stp, pwmdl tmrret sio, nvrcmd t5, #90 t5stp, pwmdl tmrret sio, #0 t5, #90 t5stp, pwmdl tmrret	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>
364 361 362 363 363 364 365 368 369 370 376 376 376 377 378 377 377 378 377 377 377 377 377	ØØD8         ØØD8           ØØD8         ØØD7           ØØD8         ØØD8           ØØD8         ØØE8           ØØE8         ØØE8           ØØF3         ØØF7           ØØF9         ØØF7           Ø1905         Ø1905           Ø1912         Ø1905           Ø1912         Ø1905	B601520C 9703D6 Version E2 (N river for NMC9 rupt Handler 835A0144AB B601501E B40151 8C2CD6 835A0144AB B601501E B401501E B401501E B401501E B401501E B40121	lov Ø2 13Ø6/93 R	t5rd9: 15:51 19 45 t5rd1: t5rd2: t5rd3:	sbit ld 87) 87) ld rbit jmpl ld ld rbit jmpl ld ld ld ld ld rbit jmpl	t5out, portpl sio, #x*93 EEPROM t5stp, pwmdl tmrret sio, nvrcmd t5, #90 t5stp, pwmdl tmrret sio, #0 t5, #90 t5stp, pwmdl tmrret nvword+1.b, sio sio, #0 t5, #90	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>
361 361 362 363 363 365 366 367 368 368 369 370 371 372 373 374 373 374 373 374 373 374 373 374 373 374 373 375 378 379 3881 3883 3885 3885 3885 3885 3885 3885	ØØD8           ØØDC           IHPC, D           IHPC, D           INPC, D           ØØDF48           ØØE8           ØØE8           ØØE8           ØØF7           ØØF40           Ø1920           Ø1941           Ø1148           Ø1188	B601520C 9703D6 Version E2 (N river for NMC9 rupt Handler 835A0144AB B601501E B40151 8C2CD6 835A0144AB B601501E B40142 9700D6 835A0144AB B601501E B40133 8CD62B 970006 835A0144AB B6015012 840121 840121	lov Ø2 13Ø6/93 R R R	t5rd9: 15:51 19 45 t5rd1: t5rd2: t5rd3: t5rd4:	sbit ld 87) 87) ld rbit jmpl ld ld rbit jmpl ld ld rbit jmpl ld ld rbit jmpl ld ld rbit	t5out, portpl sio, #x*93 EEPROM t5stp, pumdl tmrret sio, nvrcmd t5, #90 t5stp, pumdl tmrret sio, #0 t5stp, pumdl tmrret nvword+1.b, sio sio, #90 t5stp, pumdl tmrret nvword+1.b, sio t5, #90 t5stp, pumdl tmrret	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>
361 361 362 363 363 365 366 365 366 367 370 371 375 376 377 378 377 378 377 378 377 378 377 378 377 378 377 378 380 381 382 383 382 383 384 382 383 384 385 386 387 377 378 382 383 384 385 386 387 377 378 386 387 377 378 386 387 377 377 378 387 377 378 387 377 378 387 377 37	ØØD8           ØØD7           ØØD7           ØØD7           ØØD7           ØØF7           Ø102           Ø118           Ø1122           Ø122           Ø122	B601520C 9703D6 Version E2 (N river for NMCS rupt Handler 835A0144AB B601501E B40151 8C2CD6 835A0144AB B601591E B40142 9700D6 835A0144AB B601591E B40133 8CD62B 9700D6 835A0144AB B601591E B40121 8CD62A B601521C A82A AD28AB A928	lov Ø2 73Ø6/93 R R R R R R R R	t5rd9: 15:51 19 45 t5rd1: t5rd2: t5rd3: t5rd4:	sbit ld (87) (87) kar sbit jmpl ld ld ld ld rbit jmpl ld ld rbit jmpl ld ld rbit jmpl ld ld rbit jmpl ld rbit jmpl ld rbit jmpl ld rbit jmpl ld rbit st jmpl	t5out, portpl sio, #x*93 EEPROM t5stp, pwmdl tmrret sio, nvrcmd t5, #90 t5stp, pwmdl tmrret sio, #0 t5stp, pwmdl tmrret nvword+1.b, sio sio, #0 t5stp, pwmdl tmrret nvword+1.b, sio t5, #90 t5stp, pwmdl tmrret nvword+1.b, sio t5, sio, portpl A, nvword A, [nvrptr]. W nvrptr	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/097</pre>
361 361 362 363 365 366 366 366 366 366 367 368 367 378 377 378 377 378 377 378 377 378 377 378 388 38	ØØD8         ØØD7           ØØD7         ØØD7           ØØD7         ØØD7           ØØD7         ØØF0           ØØF0         ØØF0           ØØF192         ØØF0           Ø1905         Ø1907           Ø1105         Ø1105           Ø1118         Ø1118           Ø1122         Ø122           Ø122         Ø122	B601529C 9703D6 Version E2 (N river for NMCS rupt Handler 835A0144AB B601501E B40151 8C2CD6 835A0144AB B601501E B40142 9700D6 835A0144AB B601501E B40133 8CD62B 9700D6 835A0144AB B601501E B40121 8CD62A B601521C A82A AD28AB A928 8A2D	Nov Ø2 73Ø6/93 R R R R R R R R R R R	t5rd9: 15:51 19 45 t5rd1: t5rd2: t5rd3: t5rd4:	sbit ld (87) (87) kar sbit jmpl ld ld ld ld ld rbit jmpl ld ld ld rbit jmpl ld ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit jmpl l ld rbit rbit jmpl l ld rbit rbit jmpl l ld rbit rbit rbit rbit rbit rbit rbit rbit	t5out, portpl sio, #x*93 EEPROM t5, #90 t5stp, pwmdl tmrret sio, nvrcmd t5, #90 t5stp, pwmdl tmrret sio, #0 t5stp, pwmdl tmrret nvword+1.b, sio t5, #90 t5stp, pwmdl tmrret nvword+1.b, sio t5, #90 t5stp, pwmdl tmrret nvword-1, b, sio t5, #0 t5stp, pwmdl tmrret nvword-1, b, sio t5, sio t5, sio, t5, sio,	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>
3641         363           363         363           363         363           3641         365           3652         366           3653         366           3653         366           3653         366           3653         366           370         378           3773         378           380         381           381         382           3884         385           3889         3991           3923         3934           3954         3956           3978         396           3992         3935           3945         3956           3956         3978           3956         3978           3956         3978           3956         3978           3956         3978           3956         3978           3956         3978           3956         3978           3956         3978           3956         3978           3956         3978           3956         3956           3956         3956<	0008 0005 0005 0005 0005 0005 0005 0005	B601520C 9703D6 Version E2 (N river for NMCS rupt Handler 835A0144AB B601501E B40151 8C2CD6 835A0144AB B601501E B40142 970006 835A0144AB B601501E B40133 8CD62B 970006 835A0144AB B601501E B40121 8CD62A B601521C A82A A028AB A928	lov Ø2 73Ø6/93 R R R R R R R R R R	t5rd9: 15:51 19 45 t5rd1: t5rd2: t5rd3: t5rd4:	sbit ld (87) (87) (87) (87) (87) (87) (87) (87)	t5out, portpl sio, #x*93 EEPROM t5stp, pwmdl tmrret sio, nvrcmd t5, #90 t5stp, pwmdl tmrret sio, #0 t5stp, pwmdl tmrret nvword+1.b, sio sio, #0 t5stp, pwmdl tmrret nvword+1.b, sio sio, #0 t5stp, pwmdl tmrret nvword+1.b, sio sio, #0 t5stp, pwmdl tmrret nvword+1.b, sio t5, sio t5stp, pwmdl tmrret nvword-1, b, sio t5, sio t5stp, portpl A, nvvprr]. w nvrptr nvrnum	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>
361 361 362 363 363 365 366 367 366 367 372 374 3773 374 3773 3774 3773 3775 3767 3775 3776 3779 3773 3774 3773 3774 3773 3775 3776 3779 3773 3788 389 381 3823 384 385 3867 3799 3812 3823 384 385 3867 3799 3812 3823 3845 3857 3774 3772 3776 3772 3776 3779 3774 3775 3776 3779 3775 3776 3779 3774 3775 3776 3779 3788 3893 3891 3991 3992 3996 3995 3996 3996 3998 3996 3	ØØD&B           ØØD           ØØD           IHPC, J           ØØD           ØD           ØD           ØD  <	B601529C 9703D6 Version E2 (N river for NMC9 rupt Handler 835A0144AB B601501E B40151 8C2CD6 835A0144AB B601501E B40151 B401501E B401501E B401501E B401501E B401521C A82A AD28AB A928 8A2D	lov Ø2 73Ø6/93 R R R R R R R R R R R R	t5rd9: 15:51 19 45 t5rd1: t5rd2: t5rd3: t5rd4: t5rd4:	sbit ld 87) 87) ld rbit jmpl ld ld rbit jmpl ld ld rbit jmpl ld ld rbit jmpl ld ld rbit jmpl ld d ld rbit jmpl d d rbit jmpl d d d d rbit jmpl d d d d rbit jmpl d d d d d d d d d d d d d d d d d d d	t5out, portpl sio, #x*93 EEPROM t5stp, pwmdl tmrret sio, nvrcmd t5, #90 t5stp, pwmdl tmrret sio, #0 t5, #90 t5stp, pwmdl tmrret nvword+1.b, sio sio, #0 t5, #90 t5stp, pwmdl tmrret nvword+1.b, sio t5out, portpl A, nvvptr Nvrtum	<pre>; Set chip select signal to EEPROM. ; Send first part of NVR Read command. ; Format is: 1/10/A5-A0/0, TL/DD/997</pre>

C ASMHPC, Version E2 (N C-Based Driver for NMC9 mer Interrupt Handler	10V Ø2 15:5 9396/9345	1 1987)	EEPROM	195-May-88 19553 PAGE 15
404 405 0132 8601501C 406 0136 962E0F 407	t5re R	ddn: rbit sbit Here you	t5tie,pwmdl nvravl,nvrs 'll want to set a	; Yes: Terminate and pass data to CPU. ; Disable Timer T5 interrupts. ; Set NVR available for more commands. semaphore bit saying that the READ
400 409 0139 840100	;	impl	tmrret	; Return from interrupt.
410 411 0/13C 412	t5ri	nxt:		; Here, more data needs to be read from the
13 013C 97002E	R	ld inc	nvrs,#x'99	; Set up new transfer phase = Ø. : Increment address field of NVR command.
15 Ø141 892C	R	inc	nvrcmd	; (Two increments are needed: field starts in Bit 1.)
17 Ø143 962C1F	R	rbit	7,nvrcmd	Prevent increments from altering operation field. This allows addresses to roll over.
19 Ø146 9581 20		jmpl	t5rd	; Rather than triggering a Timer T5 interrupt, ; just jump to T5 Read interrupt service again.
-22 -23 Ø148	t5wi	••	: EEPR	OM Write sequence starts here.
24 0148 882E	R	ld inc	A,nvrs	; Get phase info. : Increment memory value for next 15 interrupt.
26 Ø14C 990F	'n	and	A,#x'ØF A	; Extract phase number.
28 Ø14F 29 Ø14F EC		.odd iidw		,
30 0150 1A00 0152 2A00		.ptw	t5wr0,t5wr1,t5	wr2,t5wr3
0154 3600 0156 4800				
31 Ø158 5BØØ Ø15A 69ØØ		.ptw	t5wr4,t5wr5,t5	wr6,t5wr7
Ø15C 7900 Ø15E 8800				
32 Ø16Ø 94ØØ Ø162 AØØØ		.ptw	t5wr8,t5wr9,t5	wr19,t5wr11
Ø164 AEØØ Ø166 BDØØ				
33 Ø168 C8ØØ 34		.ptw	t5wr12	
				TL/DD/99

C ASMHPC, C-Based Dr mer Interr	Version E2 (No liver for NMC93 rupt Handler	v Ø2 Ø6/93	15:51 19 45	87)	EEPROM	Ø3-May-88 10:53 PAGE 10
435 Ø16A 436 Ø16E 437 Ø171	B601520C 9701D6 835A0144AB		t5wrØ∶	sbit ld ld	t5out,portpl sio,#x'01 t5,#90	; Set chip select signal to EEPROM. ; Send start bit of EWEN command. ; Set up for interrupt at end of MICROWIRE
438 439 Ø176 440 Ø17A	B601501E 94C0			rbit jmpl	t5stp,pwmdl tmrret	; transfer. ; Start timer T5. ; Return from interrupt.
441 442 Ø17C 443 Ø17F	9730d6 835a0144ab		t5wr1:	ld ld	sio,#x'39 t5,#99	; Send body of EWEN command. ; Set up for interrupt at end of MICROWIRE transfer
445 Ø184 446 Ø188 447	B6Ø15Ø1E 94B2			rbit jmpl	t5stp,pwmdl tmrret	; Start timer T5. ; Return from interrupt.
448 Ø18A 449 Ø18E 459 Ø18E	8601521C 40 8601520C		t5wr2:	rbit nop sbit	t5out,portpl	; Remove EEPROM select momentarily to signal ; end of EWEN command, then:
451 Ø193 452 Ø196 453	9701D6 835A0144AB			ld ld	sio,#x'01 t5,#90	; Send Start Bit for ERASE command. ; Set up for interrupt at end of MICROWIRE : transfer.
454 Ø198 455 Ø19F 456	B6Ø15Ø1E 949B			rbit jmpl	t5stp,pwmdl tmrret	; Start timer T5. ; Return from interrupt.
457 Ø1A1 458 Ø1A5 459 Ø1A8	82CØ2CDA 8C2CD6 835AØ144AB	R R	t5wr3:	or ld ld	nvrcmd,#x'CØ sio,nvrcmd t5,#90	; Change NVR Command byte to ERASE command. ; Send to EEPROM. ; Set up for interrupt at end of MICROWIRE
469 461 91AD 462 91B1	B6Ø15Ø1E 9489			rbit jmpl	t5stp,pwmdl tmrret	; transfer. ; Start timer T5. ; Return from interrupt.
465 465	B6Ø1521C		t5wr4:	rbit	t5out,portpl	; Remove EEPROM chip select signal, starting ; ERASE pulse inside EEPROM.
466 Ø187 467	874E1FØ144AB			ld	t5,#TIMCON	; Set up for delay of 20 ; milliseconds (erase pulse width).
468 Ø1BD 469 Ø1C1 470	8691591E 9479			jmpl	tostp,pwmdl tmrret	; Start timer 15. ; Return from interrupt.
471 Ø1C3 472	B601520C		t5wr5:	sbit	t5out,portpl	; Set EEPROM chip select signal again, ending ; the ERASE pulse inside EEPROM.
473 Ø1C7 474 Ø1CA	9701D6 835a0144ab			ld ld	sio,#x'ø1 t5,#9ø	; Send Start bit for Write command. ; Set up for interrupt at end of MICROWIRE

TL/DD/9978-18

C ASMHPC C-Based mer Inte	C, Version E2 (No Driver for NMC93 errupt Handler	ov Ø2 306/93	15:51 19 45	87)	EEPROM	Ø3-May-88 10:53 PAGE 17
475 476 Ø10 477 Ø10 478	CF B601501E 03 9467			rbit jmpl	t5stp,pwmdl tmrret	; transfer. ; Start timer T5. ; Return from interrupt.
479 010 480 010 481 010	5 962C1F 8 8C2CD6 8 835AØ144AB	R R	t5wr6:	rbit ld ld	7,nvrcmd sio,nvrcmd t5,#90	; Create WRITE command in NVR Command byte. ; Send to EEPROM. ; Set up for interrupt at end of MICROWIRE
482 483 Ø1E 484 Ø1E 485	Ø B6Ø15Ø1E 4 9456			rbit jmpl	t5stp,pwmdl tmrret	; transfer. ; Start timer T5. ; Return from interrupt.
486 Ø1E 487 Ø1E 488	6 8C2BD6 9 835AØ144AB	R	t5wr7:	ld ld	sio,nvword+1.b t5,#90	; Send MSB of data to EEPROM. ; Set up for interrupt at end of MICROWIRE ; transfer.
489 Ø1E 490 Ø1F 491	E B601501E 2 9448			rbit jmpl	t5stp <b>,pwmdl</b> tmrret	; Start timer T5. ; Return from interrupt.
492 Ø1F 493 Ø1F 494	4 8C2AD6 7 835AØ144AB	R	t5wr8:	ld ld	sio,nvword.b t5,#90	; Send LSB of data to EEPROM. ; Set up for interrupt at end of MICROWIRE ; transfer.
495 Ø1F 496 Ø2Ø 497	C B601501E 00 943A			rbit jmpl	t5stp,pwmdl tmrret	; Start timer T5. ; Return from interrupt.
498 Ø2Ø 499 500 Ø2Ø	12 B6Ø1521C 16 874E1FØ144AB		t5wr9:	rbit ld	t5out,portpl t5,#TIMCON	; Remove EEPROM chip select, starting Write ; pulse within EEPROM. ; Set up for delay of 20
501 502 020 503 021	IC 8601501E 10 942A			rbit jmpl	t5stp <b>,pwmd</b> l tmrret	; milliseconds (write pulse width). ; Start timer T5. ; Return from interrupt.
504 505 021 506	2 B6Ø152ØC		t5wr10:	sbit	t5out,portpl	; Set EEPROM chip select signal, ending Write ; pulse within EEPROM.
507 021 508 509 021	6 9701D6 9 835A0144AB			ld ld	sio,#x'9/1 t5,#999	; Send Start bit for EWDS command (Disable ; Write/Erase). ; Set up for interrupt at end of MICROWIRE
510 511 021	E B6Ø15Ø1E			rbit impl	t5stp,pwmdl tmrret	; transfer. ; Start timer T5. . Peture from interrunt
512 022	2 59			1		, Recurr from interrupt.
512 022 513 514 022	2 59 23 970006		t5wr11:	ld	sio,#x'99	; Send body of EWDS command.
512 Ø22 513 514 Ø22	2 59 23 970006		t5wr11:	ld	sio,#x <b>'ØØ</b>	; Send body of EWDS command. TL/DD/9978-
512 022 513 514 022 C ASMHPC C-Based mer Inte	2 59 3 970006 2, Version E2 (No Driver for NMC93 errupt Handler	ov Ø2 \$Ø6∕93	t5wr11: 15:51 19 45	ld 87)	sio,#x'ØØ EEPROM	; Send body of EWDS command. TL/DD/9978 Ø3-May-88 10:53 PAGE 18
512 022 513 514 022 C ASMHPC C-Based mer Inte 515 022 516	2 59 3 970006 2, Version E2 (No Driver for NMC93 rrupt Handler 46 835A0144AB	ov Ø2 3Ø6∕93	t5wr11: 15:51 19 45	ld 87) ld	sio,#x'ØØ EEPROM t5,#90	; Send body of EWDS command. TL/DD/9978 Ø3-May-88 10:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer.
512 022 513 514 022 C ASMHPC C-Based mer Inte 515 022 516 517 022 518 022 519 022	22 59 23 970006 Driver for NNC93 Frrupt Handler 26 835A0144AB 28 8601501E 25 4C	⊃v Ø2 3Ø6/93	t5wr11: 15:51 19 45	ld 87) ld rbit jmpl	sio,#x'ØØ EEPROM t5,#90 t5stp,pwmdl tmrret	; Send body of EWDS command. TL/DD/9978 Ø3-May-88 10:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt.
512 022 513 514 022 C ASMHPC C-Based mer Inte 515 022 518 022 518 022 519 023 521 023 522 023 522 023	22 59 23 970006 27, Version E2 (No Driver for NNC93 2017 Participation Participation 26 835A0144AB 26 835A0144AB 26 835A0144AB 26 835A0144AB 26 835A0144AB 26 835A0144AB 26 835A0144AB 26 835A0144AB 27 8450 27 84500 27 845000 27 845000000000000000000000000000000000000	ov Ø2 3Ø6∕93 R	t5wr11: 15:51 19: 45 t5wr12: ;*** Hei	ld 87) ld rbit rbit rbit rbit rbit	sio,#x'99 EEPROM t5,#99 t5stp,pwmdl tmrret t5out,portpl t5tie,pwmdl t5tie,pwmdl t5tie,pwmdl uvravl,nvrs U want to set a	<pre>; Send body of EWDS command. TL/DD/9978  #33-May-88 10:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE</pre>
512 922 513 514 922 C ASMHPC C-Based mer Inte 515 922 517 922 518 922 519 923 522 923 522 923 524 923 526	22 59 23 970006 2. Version E2 (No Driver for NNC93 prrupt Handler 26 835A0144AB 28 8601501E 24 8601501E 24 8601521C 24 8601521C 28 962E0F 28 40	ov Ø2 3Ø6∕93 R	t5wr11: 15:51 19: 45 t5wr12: ;*** Hei ;*** t;	ld rbit jmpl rbit sbit re you'l ransfer jmpl	sio,#x'99 EEPROM t5,#99 t5stp,pwmdl tmrret t5out,portpl t5tie,pwmdl nvravl,nvrs ll want to set a is done. tmrret	<pre>; Send body of EWDS command. TL/DD/9978  @3-May-88 10:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE</pre>
512 922 513 514 922 C ASMHPC C-Based mer Inte 515 922 518 922 519 923 521 923 522 923 524 923 524 923 525 923 525 923 526 923 527 924 528 923 529 924	22 59 23 970006 2. Version E2 (No Driver for NNC93 rrupt Handler 26 835A0144AB 28 8601501E 24 8601501E 24 8601501C 28 962E0F 28 40 20 3E 29 528 20 3E 20 3E	ov Ø2 3Ø6∕93 R	t5wr11: 15:51 19 45 t5wr12: ;*** Hei ;*** ti tmrret:	ld rbit jmpl rbit rbit rbit re you'l ransfer jmpl pop reti	sio,#x'99 EEPROM t5,#90 t5stp,pwmdl tmrret t5tie,pwmdl t5tie,pwmdl rvravl,nvrs llwant to set a is done. tmrret psw.w A	<pre>; Send body of EWDS command. TL/DD/9978  \$3-May-88 19:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE ; Restore context.</pre>
512 922 513 514 922 C-Based mer Inte 515 922 516 922 517 922 518 922 518 922 529 923 524 923 524 923 524 923 525 923 525 923 526 923 527 924 539 924 531 924	22 59 23 970006 24 Version E2 (No Driver for NNC93 Prrupt Handler 26 835A0144AB 28 8601501E 27 46 29 6601521C 4 8601521C 4 860152	ov Ø2 3Ø6/93 R	t5wr11: 15:51 19: 45 t5wr12: ;*** Hei ;*** ti tmrret:	ld rbit jmpl rbit sbit re you" jmpl pop pop reti .end	sio,#x'99 EEPROM t5,#99 t5stp,pwmdl tmrret t5out,portpl t5tie,pwmdl nvravl,nvrs U want to set a is done. tmrret psw.w A start	<pre>; Send body of EWDS command. TL/DD/9978  \$3-May-88 19:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE ; Restore context. TL/DD/9978</pre>
512 022 513 514 022 C ASMHPC C-Based mer Inte 515 022 517 022 518 022 518 022 519 023 521 023 522 023 524 023 525 023 525 023 526 023 526 023 527 023 528 023 529 024 531 024	22 59 23 970006 2, Version E2 (No Driver for NNC93 2017 Part Handler 26 835A0144AB 28 8601501E 29 8601501C 29 8601501C 29 8601521C 29 8601521C 29 8601521C 29 8601521C 29 8601521C 29 8601521C 29 8601521C 29 8601521C 29 8601521C 20 86015	ov Ø2 3Ø6/93 R	t5wr11: 15:51 19 45 t5wr12: ;*** Hei ;*** ti tmrret:	ld rbit jmpl rbit sbit sbit rbit rbit rbit sbit sbit sbit sbit sbit sbit sbit s	sio,#x'99 EEPROM t5,#99 t5stp,pwmdl tmrret t5out,portpl t5tie,pwmdl t5tie,pwmdl t5tie,pwmdl t5tie,pwmdl t0vravi,nvrs llwant to set a is done. tmrret psw.W A start	<pre>; Return from frictropt: ; Send body of EWDS command. g3-May-88 1g:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE ; Restore context. TL/DD/9978</pre>
512 022 513 514 022 C ASMHPC C-Based mer Inte 515 022 517 022 518 022 518 022 518 022 519 023 521 023 522 023 525 023 526 023 526 023 527 023 528 023 529 024 531 024	22 59 23 970006 2, Version E2 (No Driver for NNC93 prupt Handler 26 835A0144AB 28 8601501E 29 8601501C 24 8601501C 24 8601501C 28 962E0F 28 40 29 3E 21 1	ov Ø2 3Ø6/93 R	t5wr11: 15:51 19 45 t5wr12: ;*** Hen ;*** tr tmrret:	ld rbit jmpl rbit rbit sbit rbit reyou'l ransfer jmpl pop reti .end	sio,#x'99 EEPROM t5,#99 t5stp,pwmdl tmrret t5out,portpl t5tie,pwmdl t5tie,pwmdl t5tie,pwmdl t5tie,pwmdl t5tie,pwmdl t0 set a is done. tmrret psw.W A start	<pre>; Return from frictropt: ; Send body of EWDS command. #3-May-88 10:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE ; Restore context. TL/DD/9978</pre>
512 922 513 514 922 C ASMHPC C-Based 515 922 518 922 518 922 519 923 521 93 522 923 524 923 524 923 526 923 526 923 528 923 529 924 531 924	22 59 23 970006 2, Version E2 (No Driver for NNC93 proupt Handler 26 835A0144AB 28 8601501E 27 46 29 6601521C 4 8601521C 4 860152	ov Ø2 3Ø6/93 R	t5wr11: 15:51 19 45 t5wr12: ;*** Hei ;*** ti tmrret:	ld rbit jmpl rbit rbit rbit rbit rbit rbit rbit rbit	sio,#x'99 EEPROM t5,#99 t5stp,pwmdl tmrret t5ut,portpl t5tie,pwmdl nvravl,nvrs llwant to set a is done. tmrret psw.w A start	<pre>; Retain from freerop: ; Send body of EWDS command. #3-May-88 10:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE ; Restore context. TL/DD/9976</pre>
512 022 513 514 022 C ASMHPC C-Based mer Inte 515 022 517 022 518 022 519 023 521 023 522 023 524 023 525 023 526 023 526 023 527 023 528 023 529 024 531 024	22 59 23 970006 25, Version E2 (No Driver for NNC93 25, Version E2 (No Driver for NNC93 25, Version E2 (No 835A0144AB 26 835A0144AB 28 8601501E 38 9621501E 38 962151E 38 9621501E 38 9621501E	ov Ø2 3Ø6/93 R	t5wr11: 15:51 19: 45 t5wr12: ;*** Hei ;*** ti tmrret:	ld rbit jmpl rbit sbit rbit sbit rbit jmpl pop pop pop reti .end	sio,#x'99 EEPROM t5,#90 t5stp,pwmdl tmrret t5out,portpl t5tie,pwmdl nvravl,nvrs Uwant to set a is done. tmrret psw.w A start	<pre>; Return from freerop: ; Send body of EWDS command. TL/DD/9976 g3-May-88 10:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE ; Restore context. TL/DD/9976</pre>
512 022 513 514 022 C ASMHPC C-Based mer Inte 515 022 517 022 518 022 518 022 518 022 519 023 524 023 525 023 525 023 526 023 526 023 527 023 528 023 528 023 529 024 531 024	22 59 23 970006 2, Version E2 (No Driver for NNC93 prupt Handler 26 835A0144AB 28 8601501E 27 4C 29 8601501C 24 8601501C 26 8601501C 26 8601501C 27 60 27 60 27 70 28 962E0F 28 40 27 70 28 962E0F 28 40 27 70 28 40 27 70 28 40 28 57 28 40 28 57 28 40 28 57 29 70 20 70	ov Ø2 3Ø6/93 R	t5wr11: 15:51 19: 45 t5wr12: ;*** Hen ;*** tr tmrret:	ld rbit jmpl rbit sbit rbit rbit reyou'l ransfer jmpl pop reti .end	sio,#x'\$ EEPROM t5,#99 t5stp,pwmdl tmrret t5out,portpl t5tie,pwmdl nvravl,nvrs lwant to set a is done. tmrret psw.W A start	<pre>, Retain from freerop: ; Send body of EWDS command. TL/DD/9976 g3-May-88 10:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Retror interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE ; Restore context. TL/DD/9976</pre>
512 022 513 022 514 022 C ASMHPC C-Based mer Inte 515 022 517 022 518 022 518 022 519 023 521 023 524 023 524 023 526 023 528 023 529 024 531 024	22 59 23 970006 2. Version E2 (No Driver for NNC93 Prrupt Handler 26 835A0144AB 28 8601501E 24 8601501C 28 962E0F 28 40 29 3E 21 1	ov Ø2 3Ø6/93 R	t5wr11: 15:51 19 45 t5wr12: ;*** Hei ;*** ti tmrret:	ld rbit jmpl rbit rbit rbit rbit rbit rbit rbit rbit	sio,#x'99 EEPROM t5,#99 t5stp,pwmdl tmrret t5ut,portpl t5tie,pwmdl nvravl,nvrs llwant to set a is done. tmrret psw.w A start	<pre>, Retain from frictropt: ; Send body of EWDS commend. p3-May-88 10:53 PAGE 18 ; Set up for interrupt at end of MICROWIRE ; transfer. ; Start timer T5. ; Return from interrupt. ; Remove EEPROM chip select signal. ; Disable Timer T5 interrupts. ; Set EEPROM Available. semaphore bit saying that the WRITE ; Restore context. TL/DD/9978</pre>
512 022 513 514 022 C ASMHPC C-Based mer Inte 515 022 517 022 518 022 518 022 519 023 524 023 525 023 526 023 526 023 527 023 528 023 528 023 531 024	22 59 23 970006 2, Version E2 (No Driver for NNC93 prupt Handler 26 835A0144AB 28 8601501E 27 42 29 8601501C 28 962E0F 28 40 20 3FC0 29 3E 1	ov Ø2 3Ø6/93 R	t5wr11: 15:51 19: 45 t5wr12: ;*** Hei ;*** ti tmrret:	ld rbit jmpl rbit rbit sbit rbit reyou'l ransfer jmpl pop reti .end	sio,#x'99 EEPROM t5,#99 t5stp,pwmdl tmrret t5out,portpl t5tie,pwmdl t5tie,pwmdl t5tie,pwmdl t5tie,pwmdl nvrsvl,nvrs llwant to set a is done. tmrret psw.W A start	<pre>, ketch from freerop: ; Send body of EWDS command.</pre>

ahØØC9 Abs BytealØØC8 Abs Byteb2stpØØC4 Abs Nullb8or16ØØC4 Abs Nullb8or9ØØC4 Abs NullbfunØPF4 Abs WordbfunhØPF5 Abs BytebfunhØPF4 Abs ByteblØØC0 Abs ByteblØPC7 Abs ByteblØPF4 Abs BytedirahØPF4 Abs BytedirbØPF4 Abs BytedirbØPF4 Abs BytedirbØPF4 Abs BytedirbyØPF4 Abs BytedirbyØPF4 Abs BytedirbyØPF4 Abs BytedivbyØ18F Abs	
al 00C8 Abs Byte b2stp 0007 Abs Null b8or16 0004 Abs Null b8or05 0004 Abs Null bfun 00F5 Abs Byte bfunl 00F5 Abs Byte bfund 00F5 Abs Byte bl 00CC Abs Byte dirab 00F2 Abs Note dirb 00F2 Abs Note dirb 00F2 Abs Byte dirb 00F2 Abs Byte dirb 00F2 Abs Byte dirb 00F3 Abs Null ei 0007 Abs Null	
b2stp0007AbsNullb8or160004AbsNullb8or90004AbsNullbfun0075AbsBytebfunl0075AbsBytebfunl0075AbsBytebfunl0075AbsBytebl000CAbsBytedirah0072AbsBytedirb0072AbsBytedirb0072AbsBytedirbh0072AbsBytedirbh0174AbsBytedirbh0174AbsBytedirbh0176AbsBytedirbh0187AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182AbsBytedivby1182 <td></td>	
bBor16       ØØØ Abs Null         bBor9       ØØØ Abs Null         bfun       ØØF Abs Word         bfun       ØØF Abs Byte         bfun       ØØF Abs Byte         bl       ØDC Abs Byte         dirah       ØØF Abs Word         dirbh       ØØF Abs Byte         dirbh       ØØF Abs Null         eiack       ØØØF Abs Null	
boory       ####         bfun       ###         ###       ####         bfun       ####         ###       ####################################	
blumh 00F5 Abs Byte bfumh 00F5 Abs Byte blu 00FC Abs Byte bl 00FC Abs Byte dirah 00F2 Abs Word dirbh 00F2 Abs Word dirbh 00F2 Abs Byte divby 018E Abs Byte divby 00F7 Abs Null ei 000F7 Abs Null	
bfunl       ØPFA Abs Byte         bh       ØPCC Abs Byte         bl       ØPCC Abs Byte         dirah       ØPFA Abs Word         dirb       ØPFA Abs Word         dirbh       ØPFA Abs Word         dirbh       ØPFA Abs Word         dirbh       ØPFA Abs Word         dirbh       ØPFA Abs Word         divby       Ø18F Abs Byte         doeerr       ØØ7 Abs Null         eiack       Ø92 Abs Null	
bh       ØØCD Abs Byte         bl       ØØCC Abs Byte         dirah       ØØF1 Abs Byte         dirbh       ØØF2 Abs Byte         dirbh       ØØF2 Abs Byte         dirbh       ØF2 Abs Byte         dirbh       ØF2 Abs Byte         dirbh       ØF2 Abs Byte         dirbh       Ø18E Abs Byte         dirbh       Ø1	
bl ØDCC Abs Byte dirah ØDF1 Abs Byte dirb ØDF2 Abs Word dirbh ØDF2 Abs Byte divby Ø18E Ab	
alran       ##F1 Abs Byte         dirbh       ##F2 Abs Word         dirbh       ##F2 Abs Byte         dirby       ##F2 Abs Byte         divby       #18E Abs Word         divby       #18E Abs Byte         divby	
dibh ØPF3 Abs Byte dibh ØPF2 Abs Byte diby ØF2 Abs Byte divby Ø18E Abs Word divby Ø18E Abs Byte doeerr ØØØ7 Abs Null ei ØPØ7 Abs Null eiack ØPØ2 Abs Null	
dirbl ØØF2 Abs Býte divby Ø18F Abs Word divbyh Ø18F Abs Byte divbyh Ø18F Abs Byte divbyl Ø18E Abs Byte doerr ØØØ7 Abs Null ei ØØØ7 Abs Null	
divby Ø18E Abs Word divbyh Ø18F Abs Byte divbyl Ø18E Abs Byte doerr Ø007 Abs Null ei Ø007 Abs Null	
divby pior Abs byte dover p007 Abs Null ei p007 Abs Null ejack p002 Abs Null	
doeerr 0007 Abs Null ei 0007 Abs Null eiack 0002 Abs Null	
ei	
eiack 0002 Abs Null	
eicon ØISC ADS WORD	
einole popi Abs Null	
enir ØØDØ Abs Byte	
enu Ø120 Abs Byte	
enui VI22 ADS Byte	
enur pizo nus byte eri AAAA Abs Null	
eti 0000 Abs Null	
frmerr 0006 Abs Null	
gie 0000 Abs Null	
10 9090 Abs Null	
14 0004 Abs Null	
ibuf ØØFØ Abs Byte	
ircd 0004 Abs Byte	
1rpd VVU2 ADS Byte	
lap ppp2 ADS NULL minit 006C Rel Null ROM16	
HPC-Based Driver for NMC9306/9345 Timer Interrupt Handler	PAGE 20
nvradr 0000 Abs Nuil	
nvravl 0007 Abs Null	
mundarity (10/10) Del Hand DACE	
NYFDUT ØØ2Ø KEL WOFD BASE	
nvrbyt 9924 Abs Null nvrbyt 9924 Abs Null nvrmd 922 Pal Byte BASE	
nvrby 1924 As Null nvrcnd 1922 Rel Byte BASE nvrcnd 1922 Rel Byte BASE	
nvrbyt 9020 kel word BASE nvrbyt 9004 Abs Null nvrcmd 902C Rel Byte BASE nvrdta ABCD Abs Null nvrnum 902C Rel Byte BASE	
nvrput 0020 kel word BASE nvrput 0020 kel Byte BASE nvrdta ABCD Abs Null nvrnum 0020 kel Byte BASE nvrptr 0020 kel Byte BASE	
nvrpot gø2g kel word BASE nvrpot gø2g Abs Null nvrcmd gø2c Rel Byte BASE nvrdta ABCD Abs Null nvrnum gø22 Rel Byte BASE nvrs gø22 Rel Word BASE nvrs gø26 Rel Byte BASE	
nvrbyt Ø22 Kel Word BASE nvrbyt Ø24 Abs Null nvrcmd Ø22 Rel Byte BASE nvrdta ABCD Abs Null nvrnum Ø22 Rel Byte BASE nvrpt Ø22 Rel Byte BASE nvrs Ø22 Rel Byte BASE nvrw Ø024 Rel Null ROM16	
nvrbut 0024 Abs Null nvrcmd 002C Rel Byte BASE nvrndt ABCD Abs Null nvrnum 002D Rel Byte BASE nvrptr 0022 Rel Byte BASE nvrptr 0022 Rel Byte BASE nvrwr 0022 Rel Byte BASE nvrwr 0024 Rel Byte BASE nvrwr 0024 Rel Null ROM16 nvrword 0024 Rel Null ROM16	
nvrbyt Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrdta ABCD Abs Null nvrnum Ø92D Rel Byte BASE nvrptr Ø92B Rel Word BASE nvrs Ø92E Rel Byte BASE nvrs Ø926 Asl Null nvrw Ø996 Abs Null nvrw Ø99A Rel Null ROM16 nvword Ø92A Rel Word BASE	
nvrbyt 0024 Abs Null nvrcmd 002C Rel Byte BASE nvrdta ABCD Abs Null nvrnum 002D Rel Byte BASE nvrpt 0022 Rel Byte BASE nvrum 002C Rel Byte BASE nvrum 002A Rel Null nvrn 009A Rel Null Romió nvrx 009A Rel Null Romió nvrx 009A Rel Null Romió nvvrd 002A Rel Word BASE obuf 002A Rel Word BASE	
nvrbut 0020 Kel Word BASE nvrbut 0020 Kel Byte BASE nvrdta ABCD Abs Null nvrnum 0020 Kel Byte BASE nvrpt 0022 Kel Byte BASE nvrpt 0022 Kel Byte BASE nvrw 0022 Kel Byte BASE nvrw 0024 Kel Word BASE nvrw 0024 Kel Word BASE obut 0024 Kel Word BASE 0045 Mel Abs Byte	
nvrby Ø92 Ak Null nvrcmd Ø92C Rel Byte BASE nvrndt ABCD Abs Null nvrnmu Ø92D Rel Byte BASE nvrpt Ø92E Rel Word BASE nvrs Ø92E Rel Byte BASE nvrw Ø92E Rel Byte BASE nvrw Ø92A Rel Null ROM16 nvword Ø92A Rel Null ROM16 nvword Ø92A Rel Null ROM16 portb Ø9E1 Abs Byte portb Ø9E2 Abs Word portb Ø9E2 Abs Byte	
nvrby Ø92A Abs Null nvrcmd Ø92C Rel Byte BASE nvrdt ABCD Abs Null nvrnum Ø92D Rel Byte BASE nvrpt Ø922 Rel Word BASE nvrs Ø92E Rel Word BASE nvrw Ø99A Rel Null ROM16 nvrwr Ø92A Rel Word BASE obuf Ø92A Rel Word BASE obuf Ø9EA Abs Byte portah Ø9EI Abs Byte portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte	
nvrbyt Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrdta ABCD Abs Null nvrmum Ø92D Rel Byte BASE nvrpt Ø928 Rel Word BASE nvrwr Ø928 Rel Word BASE nvrwr Ø924 Rel Null Romi6 nvrwr Ø924 Rel Null Romi6 nvword Ø924 Rel Word BASE obuf Ø9E4 Abs Byte portah Ø9E1 Abs Byte portb Ø9E2 Abs Byte portb Ø9E3 Abs Byte	
nvrbyt Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrdta ABCD Abs Null nvrrum Ø92D Rel Byte BASE nvrpt Ø928 Rel Word BASE nvrrw Ø928 Rel Word BASE nvrw Ø924 Rel Null ROM16 nvrx Ø924 Rel Null ROM16 nvrvc Ø924 Rel Word BASE obuf Ø0E1 Abs Byte porta Ø0E1 Abs Byte portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte	
nvrby Ø92 Ak Lord BASE nvrby Ø92 Ak Lord Byte BASE nvrdt ABCD Abs Null nvrnum Ø92C Rel Byte BASE nvrpt Ø92E Rel Word BASE nvrw Ø92E Rel Byte BASE nvrw Ø92E Rel Byte BASE nvrw Ø92A Rel Null ROM16 nvword Ø92A Rel Null ROM16 nvword Ø92A Rel Null ROM16 portb Ø9E2 Abs Byte porta Ø9E Abs Byte portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte portb Ø1E4 Abs Byte portd Ø194 Abs Byte portd Ø194 Abs Byte portd Ø194 Abs Byte portd Ø194 Abs Byte	
nvrbyt Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrdta ABCD Abs Null nvrnum Ø92D Rel Byte BASE nvrpt Ø92E Rel Byte BASE nvrwr Ø92E Rel Byte BASE nvrwr Ø92E Rel Byte BASE nvrwr Ø92A Rel Null Romió nvrc Ø92A Rel Null Romió nvword Ø92A Rel Word BASE obuf Ø9EØ Abs Byte portah Ø9E1 Abs Byte portb Ø4E3 Abs Byte	
nvrby Ø92 Akl word BASE nvrby Ø92 Akl Null nvrcmd Ø02C Rel Byte BASE nvrpt Ø02C Rel Byte BASE nvrpt Ø028 Rel Word BASE nvrw Ø026 Rel Byte BASE nvrw Ø026 Rel Byte BASE nvrw Ø026 Rel Nul ROM16 nvword Ø02A Rel Word BASE obuf Ø02A Rel Word BASE obuf Ø0E1 Abs Byte porta Ø0E1 Abs Byte portb Ø0E2 Abs Byte portb Ø0E2 Abs Byte portb Ø0E3 Abs Byte portb Ø152 Abs Byte portp Ø152 Abs Byte portp Ø153 Abs Byte portp Ø153 Abs Byte	
nvrby Øg04 Abs Null nvrcmd Øg2C Rel Byte BASE nvrdta ABCD Abs Null nvrnum Øg2D Rel Byte BASE nvrpt Øg2Z Rel Byte BASE nvrpt Øg2Z Rel Byte BASE nvrw Øg02 Rel Byte BASE nvrw Øg02 Rel Null ROM16 nvword Øg2A Rel Word BASE obuf Øg0EA Abs Byte portah ØgEA Abs Byte portab ØgE2 Abs Byte portb ØgE2 Abs Byte	
nvrbyt Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrdta ABCD Abs Null nvrnum Ø92D Rel Byte BASE nvrpt Ø928 Rel Word BASE nvrum Ø926 Rel Byte BASE nvrum Ø926 Rel Byte BASE nvrum Ø926 Rel Byte BASE nvrum Ø924 Rel Null ROM16 nvword Ø924 Rel Word BASE obuf Ø924 Rel Word BASE obuf Ø924 As Byte portab Ø925 As Byte portb Ø925 As Byte portb Ø925 Abs Byte portb Ø925 Abs Byte portb Ø925 Abs Byte portb Ø925 Abs Byte portb Ø953 Abs Byte portb Ø954 Abs Byte portb Ø154 Abs Byte portp Ø155 Abs Byte portp Ø155 Abs Byte	
nvrby Ø92 Akl word BASE nvrby Ø92 Akl Null nvrnum Ø92C Rel Byte BASE nvrpt Ø922 Rel Byte BASE nvrpt Ø928 Rel Word BASE nvrw Ø928 Rel Word BASE nvrw Ø924 Rel Null ROM16 nvword Ø924 Rel Word BASE obuf Ø929 Akl Null ROM16 nvword Ø924 Rel Word BASE obuf Ø9E1 Abs Byte porta Ø9E2 Abs Word portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte portb Ø9E3 Abs Byte portb Ø152 Abs Byte port Ø153 Abs Byte portp Ø153 Abs Byte port Ø154 Abs Byte port Ø155 Abs Byte port Ø156 Abs Byte port Ø157 Abs Byte port Ø157 Abs Byte port Ø157 Abs Byte port Ø157 Abs Byte port Ø159 Abs Word r1 Ø184 Abs Word	
nvrby Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrdta ABCD Abs Null nvrrum Ø92D Rel Byte BASE nvrpt Ø928 Rel Word BASE nvrrw Ø928 Rel Word BASE nvrw Ø924 Rel Null ROM16 nvword Ø924 Rel Word BASE obuf Ø9E1 Abs Byte porta Ø9E1 Abs Byte porta Ø9E2 Abs Byte portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte portb Ø9E3 Abs Byte port Ø152 Abs Byte port Ø153 Abs Byte port Ø153 Abs Byte port Ø154 Abs Byte port Ø154 Abs Byte	
nvrbyt Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrdta ABCD Abs Null nvrnum Ø92D Rel Byte BASE nvrpt Ø928 Rel Word BASE nvrum Ø926 Rel Byte BASE nvrum Ø926 Rel Byte BASE nvrum Ø926 Rel Byte BASE nvrum Ø926 Rel Null ROM16 nvword Ø924 Rel Word BASE Obuf Ø920 Abs Null porta Ø928 Rel Word BASE opurt Ø922 Abs Byte portb Ø922 Abs Byte portb Ø922 Abs Byte portb Ø923 Abs Byte portb Ø923 Abs Byte portb Ø923 Abs Byte portb Ø923 Abs Byte portb Ø953 Abs Byte portb Ø154 Abs Byte portp Ø152 Abs Byte portp Ø152 Abs Byte portp Ø153 Abs Byte portp Ø153 Abs Byte portp Ø154 Abs Word r1 Ø150 Abs Byte	
nvrby Ø92 Akl word BASE nvrby Ø92 Akl Null nvrcmd Ø92C Rel Byte BASE nvrptr Ø922 Rel Byte BASE nvrptr Ø928 Rel Word BASE nvrwr Ø924 Rel Word BASE nvrwr Ø924 Rel Word BASE nvrwr Ø924 Rel Word BASE obuf Ø925 Abs Byte porta Ø026 Abs Byte portb Ø925 Abs Byte portb Ø925 Abs Byte portb Ø953 Abs Byte portb Ø152 Abs Byte portb Ø153 Abs Byte portp Ø153 Abs Byte portp Ø153 Abs Byte portp Ø154 Abs Byte portp Ø154 Abs Byte portp Ø155 Abs Byte portp Ø155 Abs Byte portp Ø157 Abs Byte portl Ø158 Abs Byte pumdde Ø151 Abs Byte pumdde Ø159 Abs Word r1 Ø184 Abs Word r2 Ø186 Abs Word	
nvrby Ø92 Akl word BASE nvrby Ø92 Akl will nvrcmd Ø02C Rel Byte BASE nvrpt Ø022 Rel Byte BASE nvrpt Ø028 Rel Word BASE nvrw Ø026 Rel Byte BASE nvrw Ø026 Rel Word BASE nvrw Ø026 Rel Word BASE nvrw Ø027 Rel Word BASE obuf Ø027 Rel Word BASE obuf Ø028 Rel Word BASE obuf Ø028 Rel Word BASE obuf Ø028 Akl Word BASE obuf Ø028 Akl Word BASE obuf Ø028 Akl Word BASE oprta Ø028 Akl Word BASE oprtb Ø027 Akl Word BASE oprtb Ø027 Akl Word BASE oprtb Ø028 Akl Word portb Ø027 Akl Word BASE oprtb Ø028 Akl Word portb Ø028 Akl Word portb Ø153 Akl Byte powmdh Ø151 Akl Byte pwmdh Ø151 Akl Byte pwmdh Ø151 Akl Byte pwmdk Ø150 Akl Word r1 Ø184 Akl Word r2 Ø186 Akl Word r4 Ø144 Akl Word	
nvrbyt Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrdta ABCD Abs Null nvrnum Ø92D Rel Byte BASE nvrpt Ø928 Rel Word BASE nvrrw Ø926 Rel Byte BASE nvrw Ø926 Rel Byte BASE nvrw Ø926 Rel Word BASE obuf Ø927 Rel Word BASE obuf Ø927 Rel Word BASE obuf Ø928 Asel Word BASE oporta Ø928 Rel Word BASE oporta Ø928 Rel Word BASE oport Ø928 Rel Word BASE oport Ø928 Asel Word bort Ø928 Asel Word oport Ø928 Asel Word oport Ø928 Asel Word bort Ø928 Asel Word oport Ø928 Asel Word oport Ø928 Asel Word oport Ø928 Abs Byte oport Ø152 Abs Byte oport Ø152 Abs Byte oport Ø153 Abs Byte oport Ø159 Abs Byte opwmdl Ø151 Abs Byte opwmdl Ø150 Abs Word r <sup>2</sup> Ø186 Abs Word r <sup>3</sup> Ø184 Abs Word r <sup>4</sup> Ø142 Abs Word	
nvrby Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrqta ABCD Abs Null nvrnum Ø92D Rel Byte BASE nvrpt Ø928 Rel Word BASE nvrwr Ø928 Rel Word BASE nvrwr Ø924 Rel Null ROM16 nvror Ø924 Rel Null ROM16 nvord Ø924 Rel Null ROM16 nvord Ø924 Rel Word BASE obuf Ø929 Ase Null portah Ø9E1 Abs Byte portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte portb Ø9E2 Abs Byte portb Ø9E3 Abs Byte portb Ø9E3 Abs Byte portb Ø154 Abs Byte portb Ø154 Abs Byte portp Ø152 Abs Word r1 Ø158 Abs Byte portb Ø159 Abs Byte portp Ø159 Abs Byte portp Ø158 Abs Byte portp Ø158 Abs Byte portp Ø158 Abs Byte portp Ø158 Abs Byte portp Ø159 Abs Byte portp Ø159 Abs Byte port0 Ø159 Abs Byte pwmdde Ø159 Abs Word r1 Ø184 Abs Word r1 Ø184 Abs Word r3 Ø184 Abs Word r5 Ø146 Abs Word r6 Ø144 Abs Word r7 Ø146 Abs Word r6 Ø144 Abs Word r	
nvrby Ø924 Abs Null nvrcmd Ø92C Rel Byte BASE nvrpt Ø920 Rel Byte BASE nvrpt Ø928 Rel Word BASE nvrpt Ø928 Rel Word BASE nvrw Ø926 Rel Byte BASE nvrw Ø926 Rel Null Rom16 nvword Ø924 Rel Word BASE obuf Ø929 Arel Word BASE obuf Ø929 Arel Word BASE obuf Ø927 Arel Word BASE optis Ø928 Arel Word of 153 Abs Byte pwmdh Ø151 Abs Byte pwmdl Ø158 Abs Word r1 Ø184 Abs Word r2 Ø186 Abs Word r4 Ø144 Abs Word r5 Ø144 Abs Word r6 Ø144 Abs Word r7 Ø146 Abs Word	
nvrbuf 0024 Abs Null nvrcmd 002C Rel Byte BASE nvrdta ABCD Abs Null nvrrum 0025 Rel Byte BASE nvrrum 0025 Rel Byte BASE nvrrw 0026 Rel Byte BASE nvrwr 0026 Rel Word BASE nvrwr 0026 Rel Word BASE nvrwr 0026 Arel Wull ROM16 nvrwr 0024 Rel Word BASE obuf 0026 Abs Null nvrx 0024 Rel Word BASE obuf 0026 Abs Byte porta 0026 Abs Byte porta 0026 Abs Byte portb 0027 Abs Byte portb 0027 Abs Byte portb 0027 Abs Byte portf 0153 Abs Byte portp 0154 Abs Byte portp 0155 Abs Byte portp 0154 Abs Byte portp 0155 Abs Byte portp 0157 Abs Byte portp 0158 Abs Byte portp 0159 Abs Byte portp 0159 Abs Byte port0 0159 Abs Byte port0 0159 Abs Byte port0 0159 Abs Word r1 0158 Abs Word r2 0186 Abs Word r5 0146 Abs Word r5	
nvrby Ø924 Abs Null nvrcmd Ø922 Rel Byte BASE nvrdta ABCD Abs Null nvrnum Ø920 Rel Byte BASE nvrum Ø920 Rel Byte BASE nvrum Ø926 Rel Byte BASE nvrum Ø926 Rel Byte BASE nvrum Ø926 Rel Null Rom16 nvord Ø924 Rel Word BASE obuf Ø924 Rel Word BASE obuf Ø924 As Byte porta Ø924 Rel Word BASE obuf Ø924 Abs Byte portb Ø922 As Byte portb Ø922 As Byte portb Ø925 As Byte portb Ø953 Abs Byte portb Ø154 Abs Byte portb Ø155 Abs Byte portp Ø155 Abs Byte portd Ø159 Abs Word r1 Ø159 Abs Word r1 Ø159 Abs Word r1 Ø184 Abs Word r5 Ø144 Abs Word r5 Ø146 Abs Word r5 Ø147 Word r5 Ø147 Word r5	

NSC ASMHPC, Version E HPC-Based Driver for I Timer Interrupt Handle	2 (Nov Ø2 15:51 1987) MC93Ø6/9345 er	EEPROM	Ø3-May-88 10:53 PAGE 21
sk         ØØØ6         Abs         Null           snvr1         ØØ32         Ret         Null           snvr1         ØØ32         Ret         Null           snvr1         ØØ32         Ret         Null           sram         ØØØ7         Ret         Null           sram11         ØØØ7         Ret         Null           sram12         ØØ12         Ret         Null           start         ØØØØ         Ret         Wolf           suwlp         Ø12         Ret         Null           suwlp         Ø920         Ret         Null           type         Ø12         Ret         Null           type         Ø12         Ret         Null           suwlp         Ø20         Ret         Null           type         Ø12         Abs         Null           type         Ø12         Abs         Null           type         Ø12         Abs         Null           type         Ø12         Abs         Null           type         Ø18         Abs         Null           type         Ø12         Abs         Null           type	ROM16 ROM16 ROM16 BASE ROM16 ROM16 ROM16 ROM16 ROM16 ROM16		
			TL/DD/9978-23
NSC ASMHPC, Version E2 HPC-Based Driver for N	2 (Nov Ø2 15:51 1987) MC93Ø6/9345	EEPROM	TL/DD/9978-23 Ø3-May-88 10:53 PAGE 22
NSC ASMHPC, Version E2 HPC-Based Driver for M Timer Interrupt Handle t5 Ø144 Abs Worc t5ack ØØØ7 Abs Null t5int ØØB8 Rel Null t5int ØØB8 Rel Null t5poll ØØB2 Rel Null t5rdØ ØØD8 Rel Null t5rdØ ØØD8 Rel Null t5rdØ ØØD8 Rel Null t5rdØ ØØB8 Rel Null t5rdØ ØØFA Rel Null t5rdØ Ø120 Rel Null t5rdØ Ø132 Rel Null t5rdØ Ø132 Rel Null t5rdØ Ø164 Rel Null t5rdØ Ø165 Rel Null t6rdØ Ø165 Rel Null	2 (Nov Ø2 15:51 1987) IMC9386/9345 Fr ROM16 ROM1	EEPROM	TL/DD/9978-23

NSC ASMH HPC-Based Timer Int	PC, Version E2 (Nov Ø2 15:51 d Driver for NMC93Ø6/9345 terrupt Handler	1987)	EEPROM	Ø3-May-88 1Ø:53 PAGE 23
t7ack t7out t7out t7out t7out t7out t7out t7out t7out toout	0007 Abs Null           0007 Abs Null           0005 Abs Null           0005 Abs Null           0006 Abs Null           0007 Abs Byte           0107 Abs Byte           0109 Abs Word           0042 Rel Null ROM16           023C Rel Null ROM16           023C Rel Null ROM16           0206 Abs Null           0006 Abs Null           0006 Abs Null           0006 Abs Null           0008 Abs Null           0007 Abs Byte           0007 Abs Null           0008 Abs Null           0007 Abs Byte           0007 Abs Byte           0007 Abs Byte           0076 Abs Null           0087 Abs Null           0087 Abs Null           00885 Abs Null           0087			
**** Er	rors: Ø, Warnings: Ø			TL/DD/9978-25

Interfacing A Serial EEPROM to the National HPC16083

# 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:

 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.  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 does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.