How To See the COP Watchdog Fire in the Nohau Trace Buffer with Motorola HCS12 Microcontrollers

Application Note *by Doron Fael & Robert Boys* **V2.6 January 16, 2002**

Purpose

This note demonstrates the Nohau full emulator's ability to operate through a COP Watchdog or a user RESET sequence and record the associated instructions in the trace memory. This ability is essential for fast debugging and operational code integrity in the critical area of Watchdog service routines. This note is for the Motorola HCS12 family but is also applicable for the standard HC12 family such as the DG128A. These are covered in a separate application note.

This feature of the emulator allows you to determine such things as where the program was when the COP fired, how much time elapsed between COP refreshes to ARMCOP, how often the COP fires, what was executed after the COP vector was fetched, any spurious writes to ARMCOP and many others. The techniques described are also useful for interrupt and RESET debugging.

COP Watchdog for the HCS12 Microcontrollers

The HCS12 microcontrollers have a COP Watchdog timer used to recover from programming errors. The most likely scenario is one where the CPU has entered a loop with a condition that is never met. If set, the COP will time-out and reset the CPU and restart at the address specified by the COP Watchdog vector at FFFA.

Barracuda I and II Chips (I mask is OK36N and II mask is OK79X) PC9S12DP256

The Barracuda I chip will cause a system reset if the COP is enabled and the BDM becomes active. See Errata MUCTS00362. The workaround is to debug with the COP disabled. The result to the BDM and full emulators for Barracuda I is that the COP cannot be activated and used for debugging purposes. This application note will not work with mask set OK36N. The reason is that if the COP is enabled, it will cause a Reset as soon as emulation is stopped and the emulator Monitor Mode becomes active because the BDM is then activated. The emulator enters its Monitor Mode when STOP is clicked and emulation stops.

The COP on all Barracuda I full emulators has been automatically disabled and the user is not able to enable it. The COP is disabled by the emulator writing to the COPCTL register right after every Reset using our Reset Transition Code. Nohau no longer uses the OK36N mask set. Contact Nohau to have this chip updated. If you disable the Reset transition in the software under Config, Emulator, you will bypass the Nohau startup code and go directly to your own code but with certain restrictions. The Barracuda COP will work correctly but there are restrictions such as no breakpoints in the first 1000 CPU cycles. See the hardware manual for details.

The Barracuda II (OK79X) does not have this effect. You must have this mask or a later version, another family member or a "H" device (PC9S12H256 family) in order for the demonstration to work properly.

A new Bit RSBCK - Bit 6 was added to the COPCTL register (address 003C) on the Barracuda II. When set, this bit stops the COP Watchdog and the RTI when the BDM becomes active. This happens when the emulator goes into Monitor mode. The COP and RTI counting resumes when emulation is on by clicking on GO. This bit can be written anytime in special mode but only once in user mode but if no special write has already occurred.

Registers

COPCTL Watchdog Register

The COPCTL register (address 003C) is 'Write Once' after every Reset. The emulator must set the RSBCK bit (bit 6) by writing to the COPCTL register right after every Reset in the emulator Reset Transition code. This action uses up the one allowable write with the result customers are not able to write this register in their code. Any subsequent writes to the COPCTL register by their code are ignored.

To overcome this problem the customers need to manually specify their required COPCTL value under the Config Emulator Misc Setup tab, RCRCTL Register Write Value field. The COP is activated when any of bits 0 through 2 are set to a one. Three zeroes deactivate it. These three bits specify one of seven time-out rates for the COP. The emulator start-up code, also called the Transition Code, will place this value into the COPCTL register after ORing it with 40 to set the RSBCK bit.

The COP Watchdog default is disabled after Reset and the only method to enable it is to enter the appropriate value into bits 0 through 2 in the RCRCTL field. RCRCTL is not a register in the HC12 microcontroller but a Nohau register name. This will be changed in the software to read COPCTL. The Transition code can be deactivated by the Seehau software to allow the user code to start directly after a hardware or COP reset. Call Nohau technical support for details if you need this feature.

The values which may be specified in the Nohau RCRCTL field are:

40	to disable the COP.
41 through **47**	to enable the COP at various time-out periods. (41 is used in this note).

Bit 7 may be set for Windowed COP operation (to force both minimum and maximum COP time-out periods). All writes to the ARMCOP must be within the last 25% of the selected time period.

This register can be displayed in a register window but since it has already been changed once by the Nohau Transition (start-up) code, it is read only.

ARMCOP Watchdog Register

To prevent COP time-outs, writes of 55 and AA (in this order only) to the ARMCOP register at address 003F are required periodically to prevent the COP Watchdog from firing and causing a reset and fetch of the COP vector at address FFFA.

This register can be viewed in the Seehau register window and can be written to but will always return zeroes as defined by Motorola. If you enter any values other than those specified by Motorola (55 and AA) or out of sequence, the COP will fire immediately. The Nohau Shadow memory in either a data or register window will reflect the writes in real time.

COP Watchdog Demonstration

COP Watchdog Demonstration

The Example Program

The example program COPRST.HEX is in the C:\Nohau\Seehau12\Examples\DP256\COP directory if Seehau is installed using the default paths. Contact Nohau tech support if you do not have this file. Assembly Source files are included.

Figure 1 shows the part of this program of interest here.

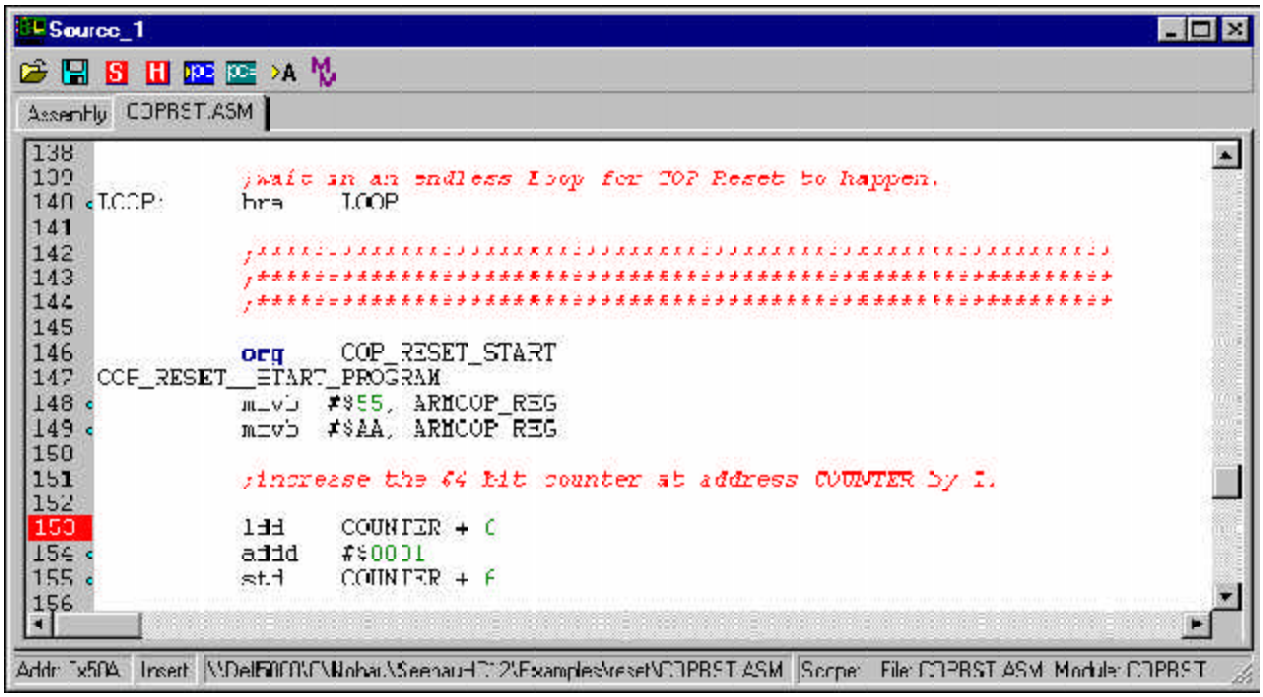


Figure 1

The program starts at address 400 and the emulator program counter will point to this address after reset. The program will do the regular initialization procedures including clearing an 8 byte counter at 7F0. It will then wait at Line 140 (Loop) in an endless loop waiting for a COP Time-out Reset to occur.

When the COP Reset occurs, the CPU is Reset, and operation is resumed at address 500 (Line 147) as specified by the COP_RESET_VECTOR (at address FFFA). The ARMCOP register will be refreshed with 55 and AA at Lines 148 and 149 which the method used to keep the COP from firing. The program then increments the 8 bit counter at 7F0 by 1, and jumps to CONTINUE at Line 81 (not shown here) and will again come to the bra LOOP at Line 140 as before to wait for another COP watchdog Reset to occur. This will happen because ARMCOP is not being refreshed at this point. These actions including the RESET sequence and vector fetching will all be recorded in the timestamped trace memory.

Many COP Watchdog Resets will occur and the counter at address 7F0 will increment by 1 every time a COP Reset occurs. Most of the incrementing of the counter will be at 7F6 which is the least significant word of COUNTER. This will be visible in real-time with the Nohau Shadow RAM.

COP Watchdog Demonstration

Configuring The Software

You can configure the software with the following instructions or use the macro *cop.bas* that is included in the same directory as the example program.

1) In the Config, Emulator window check the Disable EEPROM and Enable COP Watchdog boxes. Leave all others set to the default values. Disable EEPROM substitutes emulator RAM for the EEPROM so it is easily written to without programming algorithms. Default EEPROM starts at 400.

2) Select Normal Single Chip mode in the same window. Leave all else at the default values.

3) Add the DDRE, COPCTL and ARMCOP registers to the register window if necessary. Do this by right mouse clicking on the register window (Reg_1) and selecting Add Special Register (SFR). In the SFR window that opens in turn select DDRE (under I/O Control), COPCTL and ARMCOP (both under CRG) pressing the ">" button to copy the items over to the right. When this is completed, select the ADD To... button and click on Reg_1. These registers will be added to the register 1 window. You may make a new window if you prefer.

4) Click on DDRE in the register window and then right mouse click. Select Change Attributes and check the Enable Reset Value box. Enter 60 in the Reset Value box. This will set DDRE to 60 when the chip is reset. This is needed because the emulator is working in stand-alone mode and there is no pullup resistance that would be provided by standard target systems.

5) Under the Config, Emulator, Misc Setup Tab, set the RCRCTL Register Write Value type in 41 to enable the COP Watchdog. This box will change to COPCTL.

6) Load the file COPRST.HEX.

7) Place a breakpoint at line 153 in the COP_RESET_START_PROGRAM routine by clicking on the line number once.

8) Open a Data window and point it to address 7F0. Select Shadow and 16 bit display at the bottom of this window.

9) Configure the Trace under Config, Trace, Includes for Triggers and Filters, and check the filter items: Instruction Execution, CPU Reads and Writes, Reset State and Reset Transition State.

10) Select Config, Save Settings to save your configuration. Use startup.bas as the default or a macro of your own. Check the box under Config, Environment box labeled "Use Startup Dialog?" in order to be asked at Seehau startup which macro file you want to use. See below under "Using Seehau Macros ..." for more information.

11) Click on the Go button.

12) The program will stop on line 153 - address 50A where the breakpoint is set. The trace will show the history including the Reset State and the Reset Transition State followed by the COP Reset vector fetch from address FFFA, and the execution start at address 500. This is shown in Figure 2.

Using a Seehau Macro for the Settings

The above settings are contained in two Nohau macros contained in the Examples/DP256/COP directory. They are named COPdp256.bas for the DP256 family and COPh256.bas for the H256 family. Numerous windows and settings will be automatically installed on your Seehau session. You must check the box labeled "Use Startup Dialog?" in the Config, Environment window. You could also place the macro name in the c:\Nohau\SeehauHC12\seehau.ini file in the appropriate line.

COP Watchdog Demonstration

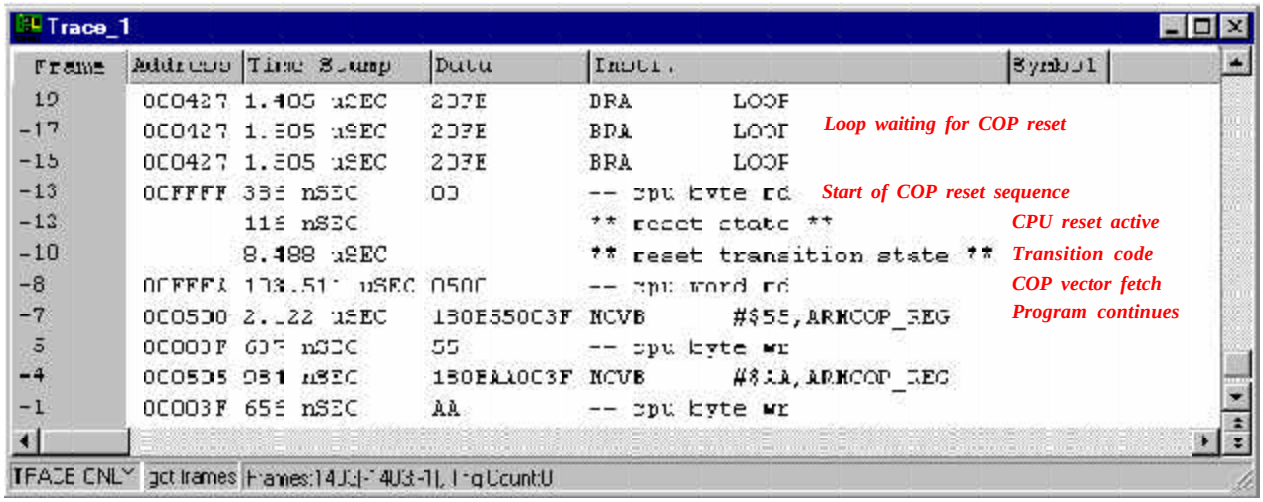


Figure 2

Trace Explanation

The Source window shown in Figure 1 displays the code as it is written. All code is shown with labels and comments. The assembly language can be shown with the debugging information as in Figure 1 or with disassembled assembly code by clicking on the Assembly tab. If this program was written in C, this would be shown also. The C source and assembly can be displayed together using Mixed mode.

The trace window shows the instructions as they were actually executed. Instructions not executed because of jumps or branches will not show in the trace as they will in the Source window. C source can also be displayed in the trace window if it is present.

1) Referring to Figure 2, Frames 19 through 15 show the endless loop branch instruction. Note the times this was happening. Embedded in here are Free Cycles and Fetch cycles. Select Config, Trace and in the Includes box select Free Cycles and Fetch cycles in the Filter section (after the Trigger section).

2) Frame 13 is the start of the CPU RESET sequence as the COP Watchdog fires.

3) Frame 12 is the CPU exiting the RESET state.

4) Frame 10 is the emulator entering its Reset Transition State or start-up code. This part can be disabled (but not without consequences) by unchecking the Reset Transition box in the Config, Emulator menu.

5) Frame 8 is the COP Vector fetch at FFFA. The value fetched is 0500 and is a CPU word read.

6) Frame 7 is the start of the execution at address 500 as specified by the COP vector. You can see the MOVB instruction at Frame 7 with the associated byte write of 55 to address 3F which is the ARMCOP register.

7) Frame 4 and 1 is the write of AA to the ARMCOP register. This is where the program execution was stopped by the breakpoint set at Line 153 (address 50A). This is one instruction past the MOVB at 505 but it was not executed because Nohau breakpoints are no-skid. The instruction a breakpoint is set to is not executed. If this breakpoint was not set, the program execution would continue forever. The counter at 7F6 will increment and the trace can be viewed without stopping the emulation by clicking on the Trace icon in the main menu.

COP Watchdog Demonstration

Some more interesting items...

1) Click on the GO button a few more times and watch the counter at 7F6 incrementing.

2) Remove the breakpoint at line 153, and again watch the counter as it increments every time a COP Reset is detected. Stop the trace by clicking on the trace icon and look at the history. It will be full of BRA LOOP instructions because this is the most executed instruction.

3) In order to prevent the BRA LOOP instruction at 427 from filling the trace, open the Config, Trace window, click on the Filter tab and right click on the address field and enter the values of 0 - 426 and 429 - FFFF as shown in Figure 3. Use either the Include All or the Instruction Execute fields which will appear. Do not enter any Data fields. Click OK. You will have to start and stop the trace to put your changes in effect. If you use the provided macro, these values are already entered. Just click on the Filter checkbox in the Trace Setup tab to activate the Filter mechanism.

4) Stop the trace and you will see the history recorded in the trace without the BRA LOOP instructions. Note that the Reset State and Reset Transition State trace frames don't show, since a trace filter is set.

5) Write 40 to the RCRCTL Register Write Value to disable the COP watchdog when the emulation is stopped. Click on GO and note the program never leaves the BRA LOOP instruction since the COP is now disabled. The counter at 7F6 is never incremented.

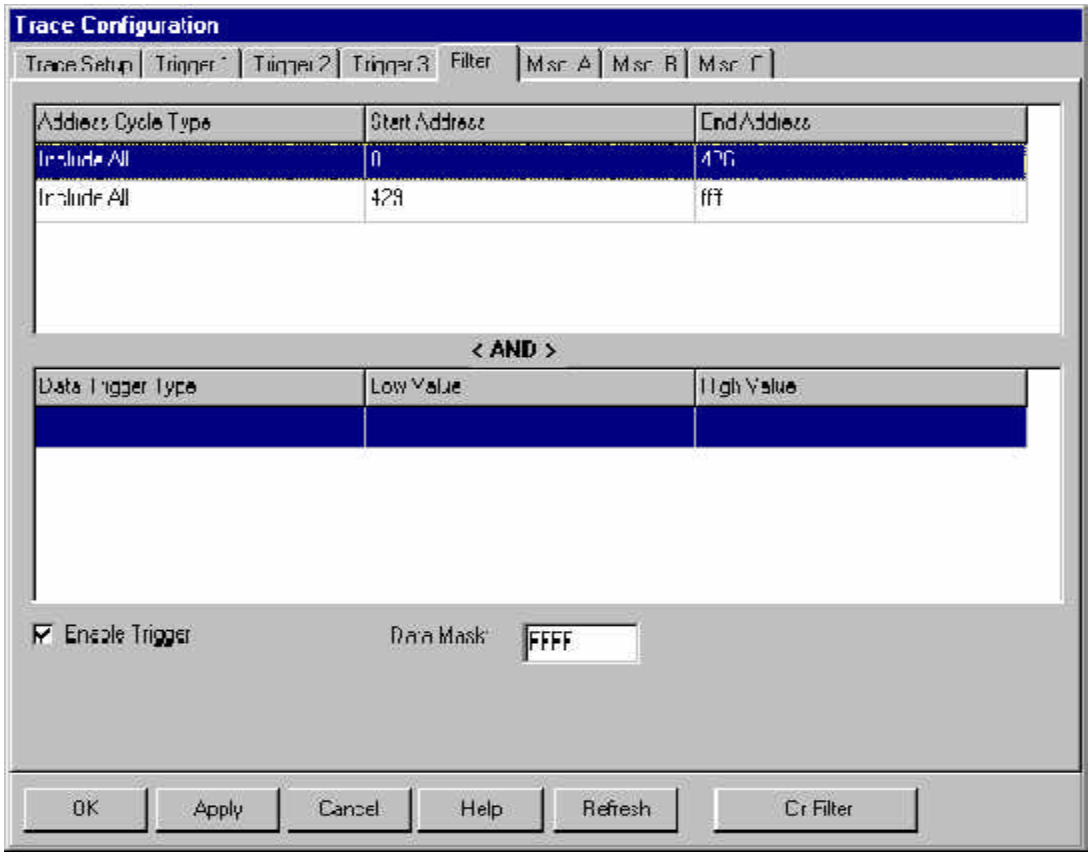


Figure 3

Target RESET Demonstration

Target RESET Demonstration

Background

If the target system asserts the RESET line AND the RESET From Target box in Config, Emulator is checked, the CPU on the emulator will be reset. This event can be tracked in the Nohau trace memory and the emulator will operate properly through this sequence. This feature is very useful for debugging RESET sequences similar to the COP Watchdog and is also very useful for interrupt problems.

Hardware Modification:

Add a 4.7 Kohm pullup resistor from the RESET pin on the emulator header to + 5 volts. The VCC-TP pin on the emulator body is a good place to obtain 5 volts. Be careful of using the VDD supplies on the personality module as some of these voltages are turned off by the emulator when it is in monitor mode or at startup. This pullup will normally be supplied by the target system.

This demonstration will use the same program as the COP reset with the COP disabled. You can use the same macros or instructions from the preceding section but these instructions must be followed:

1) To disable the COP write 40 to RCRCTL Register Write Value in the Config, Emulator, Misc Setup tab. Recall RCRCTL will change to the correct name COPCTL. You can also uncheck the Enable COP Watchdog box to disable the COP.

2) Check the Reset From Target check box in the Config, Emulator window to allow a Reset from the target to be recognized by the emulator.

3) Remove the breakpoint at Line 153 and place another on Line 63 of the program.

5) Uncheck the Filter in the Config, Trace window to disable the Filter and all instructions will be recorded.

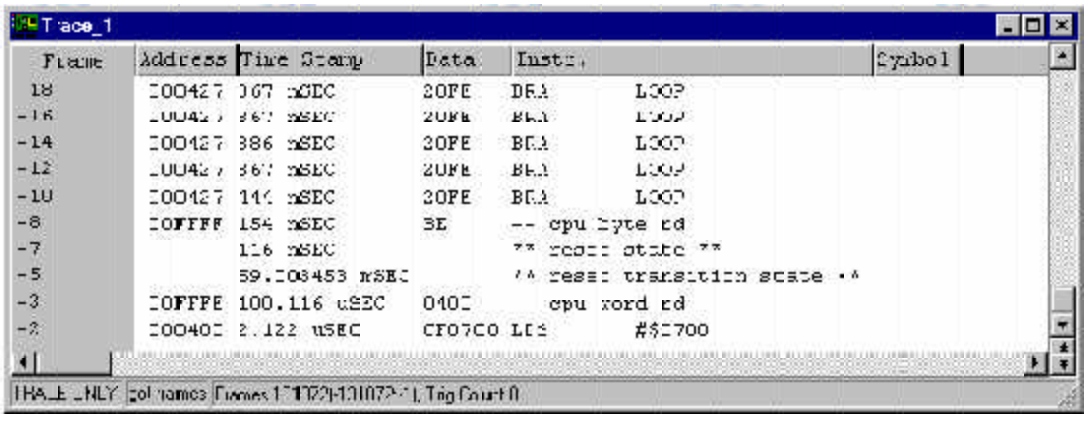
6) Run the program by clicking on GO and the breakpoint will be met. If it does not stop here, the COP is probably still operating. (Recheck your settings as mentioned in paragraph 1).

7) Click on the Go Button again. Now the program does not stop and is stuck at the BRA LOOP endless loop. It is waiting for a COP reset that will never occur because you have disabled the COP. You can stop the emulation temporarily to confirm this by the BRA LOOP instructions recorded in the trace.

8) Briefly short the Reset signal on the CPU-Module to GND for a short duration. The RESET LED will blink and the program will stop on the breakpoint at Line 63 at address 403.

9) The trace will show the history including the Reset State, Reset Transition state (Nohau startup code), and the Reset Vector Fetch from address FFFE on Frame 3 as shown below in Figure 4.

Figure 4



To locate your local representative go to www.nohau.com/reps

Nohau **Tel: (888) 886-6428**
275 E. Hacienda Avenue Tel: (408) 866-1820
Campbell, California 95008 Fax: (408) 378-7869
Email: sales@nohau.com Web: www.nohau.com

Enable Engineering **Tel: (800) 68-NOHAU**
430 Peninsula Ave #14 Tel: (650) 375-0409
San Mateo, CA 94401 Fax: (650) 375-8666
Email: sales@eecosales.com Web:ww.eecosales.com

Dearborn Group **Tel: (248) 488-2080**
27007 Hills Tech Ct.
Farmington Hills, MI 48331 Fax: (248) 488-2082
Email: dg@dgtech.com Web: www.dgtech.com

Nohau UK Ltd. **Tel: +44 1962 733 140**
The Station Mill Fax: +44 1962 735 408
Alresford, Hampshire S024 9JG
Email: sales@nohau.co.uk Web: www.nohau.co.uk

MICROTASK Embedded S.r.l. **Tel: +39 02 6680 2557**
Via Paolo Bassi, 1 Fax: +39 02 6900 8174
20159 Milano
Email: mcavallaro@microtask-embedded.it

Nohau Elektronik GmbH **Tel: +49 (0) 7043/9247-0**
Goethestr. 4 Fax: +49 (0) 7043/9247-18
75433 Maulbronn
Email: sales@nohau.de Web: www.nohau.de

Nohau Elektronik AB **Tel: +46 40 59 22 00**
Derbyvägen 4 Fax: +46 40 59 22 29
S-212 35 Malmö, Sweden
Email: info@nohau.se Web: www.nohau.se

Emulations S.A.R.L. **Tel: +33 1 69 41 28 01**
A13 - Burospace Fax: +33 1 60 19 29 50
Chemin de Gizy
91572 Bievres Cedex, France
Email: sales@nohau.co.uk Web: www.nohau.co.uk

Microcomputer Solutions Pvt. Ltd.
Tel: +91 20 4210615/4217725
22/6, Premnagar Fax: +91 20 4216798
Pune 411 037, India
Email: mspls@vsnl.com Web: www.mspls.com

Sophia Systems Co., Ltd. **Tel: +81 44 989 7239**
6-2, Minamikurokawa, Asao-ku Fax: +81 44 989 7005
Kawasaki-city, Kanagawa 215-8588, Japan
Email: TSC@sophia-systems.co.jp
Web: www.sophia-systems.co.jp