Getting Started with the Nohau EMUL51-PC Emulator

For the µPSD32xx Family of Microcontrollers

February 2003

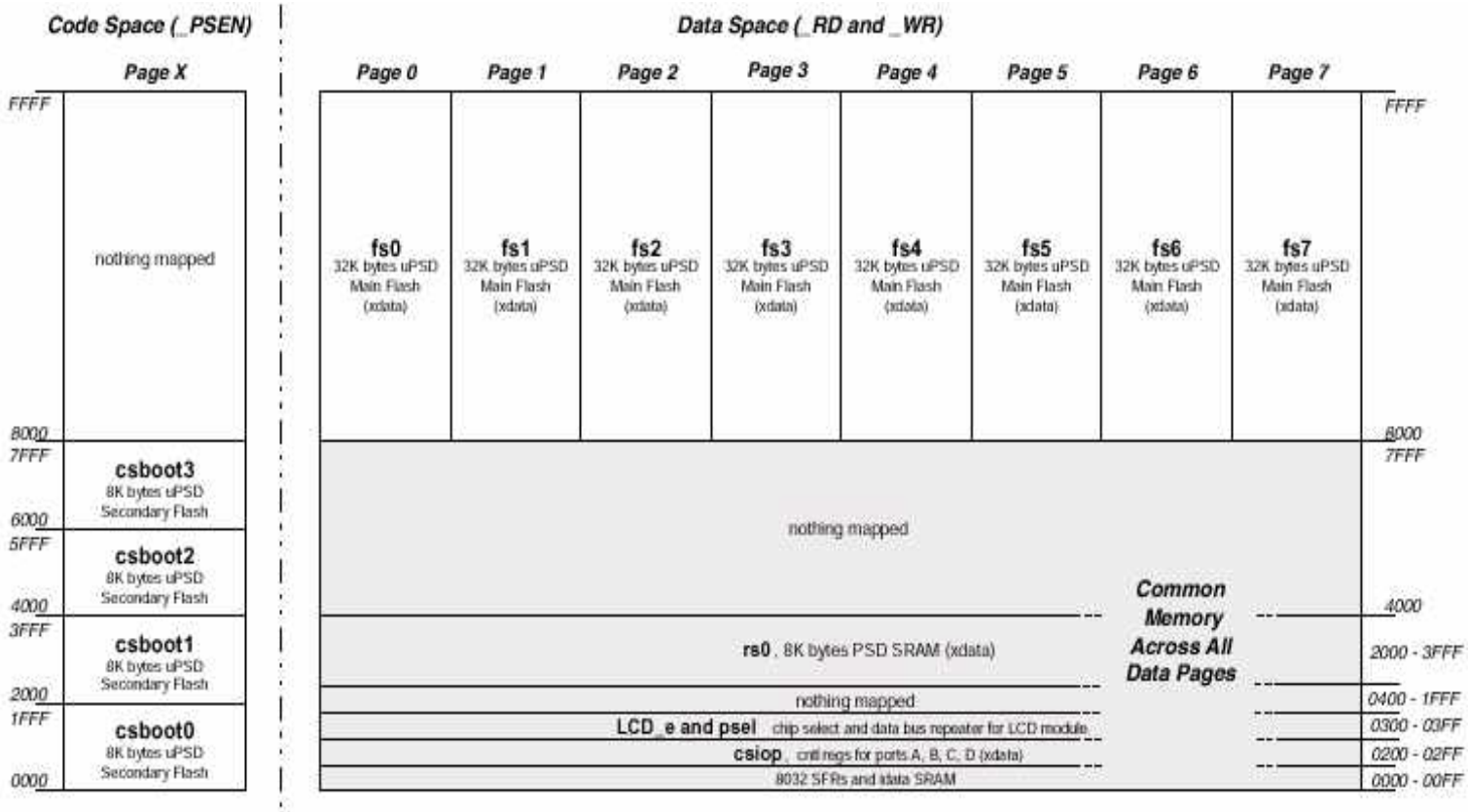
Introduction

The main purpose of this application note is to help a uPSD customer get started debugging with the Nohau Emul51-PC Emulator for the uPSD 3200 microcontroller family from STMicroelectronics.

IMPORTANT: Because the uPSD has programmable decode logic that allows the user to map the Flash and SRAM memory to different addresses in the 8032 address space, it is critical that all Development Tools use the SAME memory map definition. This includes the complier used to generate the code for the uPSD (Keil or some other), the uPSD itself (via PSDSoft Express) and the Nohau emulator.

First, the user must define (select) the memory map model that they would like to use for the intended application. This might be a simple flat memory map (no paging) or may be a more complex memory map with paging or swapping. The design of the memory map is critical to take advantage of the various features of the uPSD such as paging, multi bank flash and swapping flash into XRAM. After the memory map is selected, all the development tools must be configured to use the same memory map.

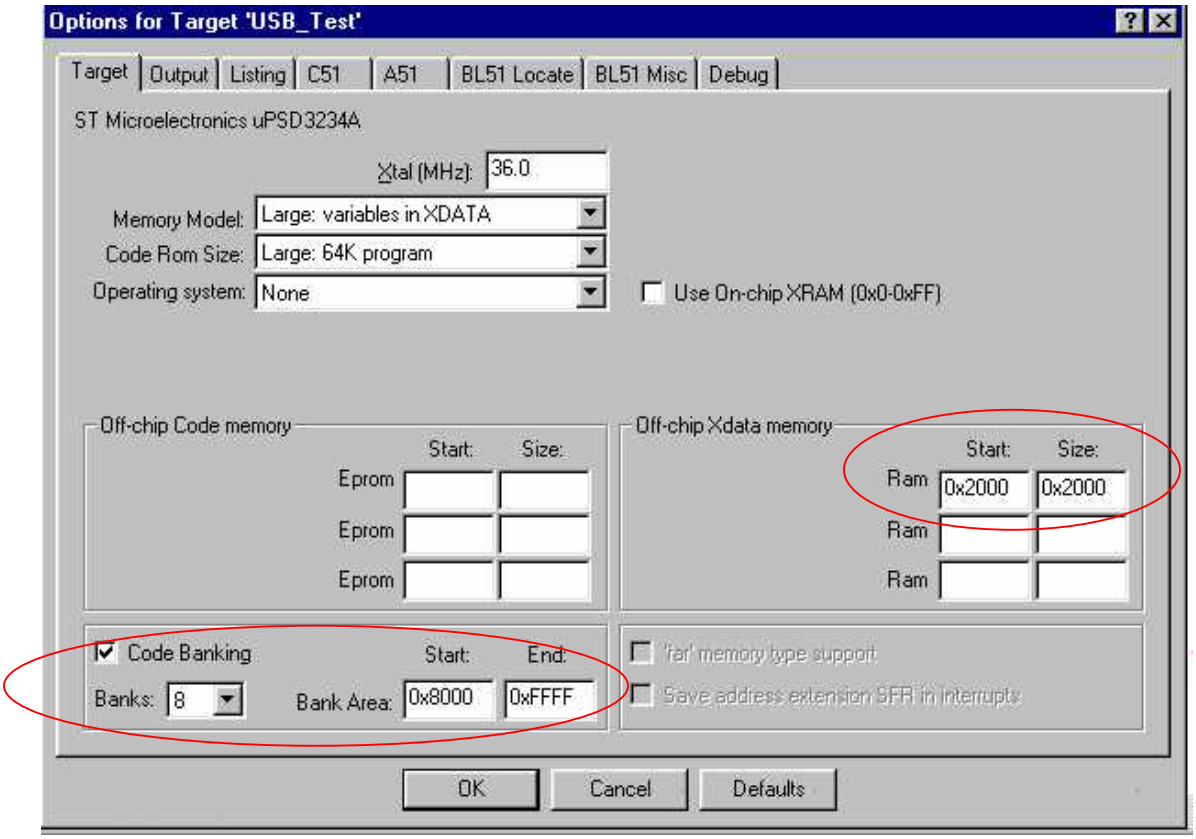
For demonstration purposes, all of the configuration information in this application note will be based on the design example given in the application note AN1560. Please see AN1560 for more details. A summary of the memory map in AN1560 is:



Configuration Setup for Keil uVision2

After the memory map is defined, the user must describe the memory map to the compiler or IDE environment. This must be done BEFORE any code can be compiled or linked by the programming tools. How and where you map the code space (Flash) and where you locate the XRAM (SRAM) affects how code is generated for the 8032 environment. In addition, if you are going to use paging, then this must be configured in the compiler to match your memory map definition in order for it to work properly.

The following is an example for setting up the XRAM address location for the Keil IDE environment. The 'Options for Target' for the Keil complier needs to be setup up as follow:



Note the Off-chip Xdata memory RAM starting address is set to 0x2000 and it's size is set to 0x2000 (8KB). It is setup this way because the example in AN1560 mapped the uPSD's SRAM to this location. Also note that the Code Banking is checked, with Banks set to 8. This example has the main flash setup to use 8 pages, with address range of 0x8000 – 0xFFFF for each page. Based on the value in the page register (0-7), a sector of the main flash will be mapped to code space 8000 to FFFF.

Configuration Setup for uPSD

IMPORTANT: The memory map that you defined must be configured into the uPSD using the PSDSoft Express configuration tool provided by STMicroelectronics. Every uPSD must be configured before the device will work. There is NO default memory map for the uPSD.

The uPSD features a programmable decode PLD that is used to decode and map the Flash Sectors, SRAM and I/O space into the 8032 address space. Recall that the 8032 has a separate Code space of 64KB and a separate XDATA space of 64KB.

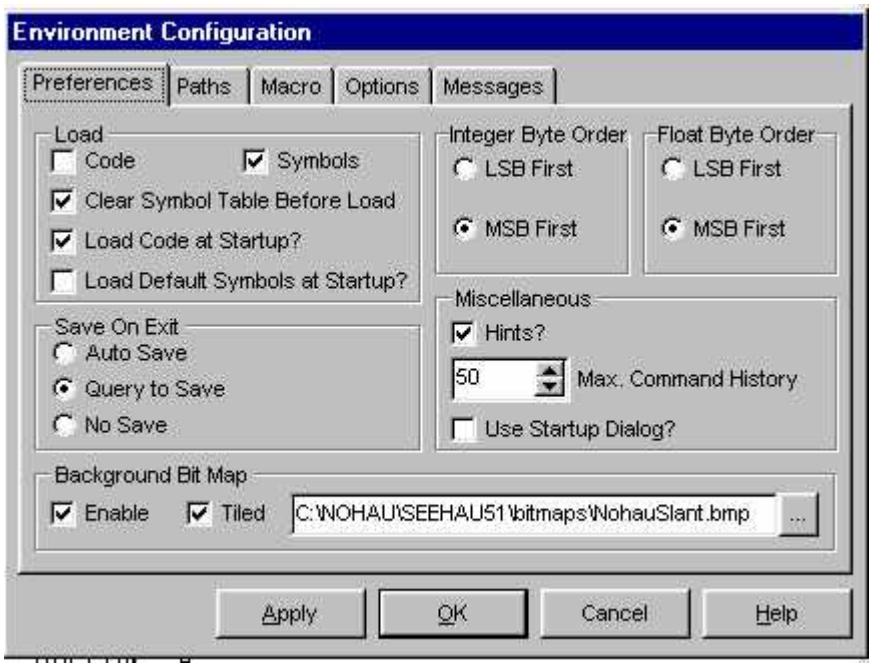
The four memory blocks (Main flash, Boot flash, SRAM, and the CSIOP Space) of the uPSD are external to the 8032 core and are selected when 8032 addresses are presented to the uPSD's Decode PLD. Using PSDsoft Express, the user will define the memory map that they have selected using the windows based GUI. This configuration information will be used to automatically program the Decode PLD in the uPSD. The user will define items such as the starting address of each flash segment, where the SRAM is located and the page register configuration. The memory map must match the address locations used in source code. For example, the I/O space register (CSIOP) is used to point to a block of configuration registers inside the uPSD. The CSIOP base address is assigned to address 0x200 in the source code, and has a length of 0x100 (256 bytes). In the PSDsoft Express, we need to assign CSIOP to this address range so that the source code will be able to access the uPSD's registers starting at address 200 in the XDATA address space. The same is true for the uPSD's SRAM chip select (RS0), which should be mapped to 0x2000 – 0x3FFF. FS0-7 are the main flash sectors, CSBoot0-3 are the boot sectors. The memory map should look like the following:

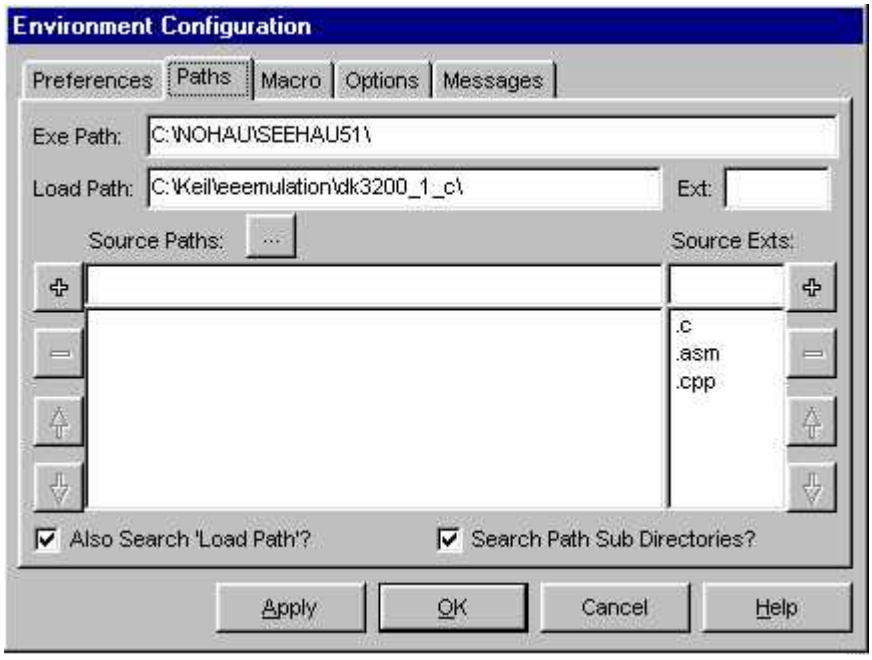
Label	Page Bit	start	end	Mem
y=page(don't	care)			
fs0	Obyyyyy000	0x8000	0xFFFF	F
fs1	Obyyyyy001	0x8000	0xFFFF	F
fs2	Obyyyyy010	0x8000	0xFFFF	F
fs3	Obyyyyy011	0x8000	0xFFFF	F
fs4	Obyyyyy100	0x8000	0xFFFF	F
fs5	Obyyyyy101	0x8000	0xFFFF	F
fs6	Obyyyyy110	0x8000	0xFFFF	F
fs7	Obyyyyy111	0x8000	0xFFFF	F
csboot0	Obyyyyyyyy	0x0000	0x1FFF	F
csboot1	Obyyyyyyyy	0x2000	0x3FFF	F
csboot2	Obyyyyyyyy	0x4000	0x5FFF	F
csboot3	Obyyyyyyyy	0x6000	0x7FFF	F
rs0	Obyyyyyyyy	0x2000	0x3FFF	V
csiop	Obyyyyyyyy	0x0200	0x02FF	V

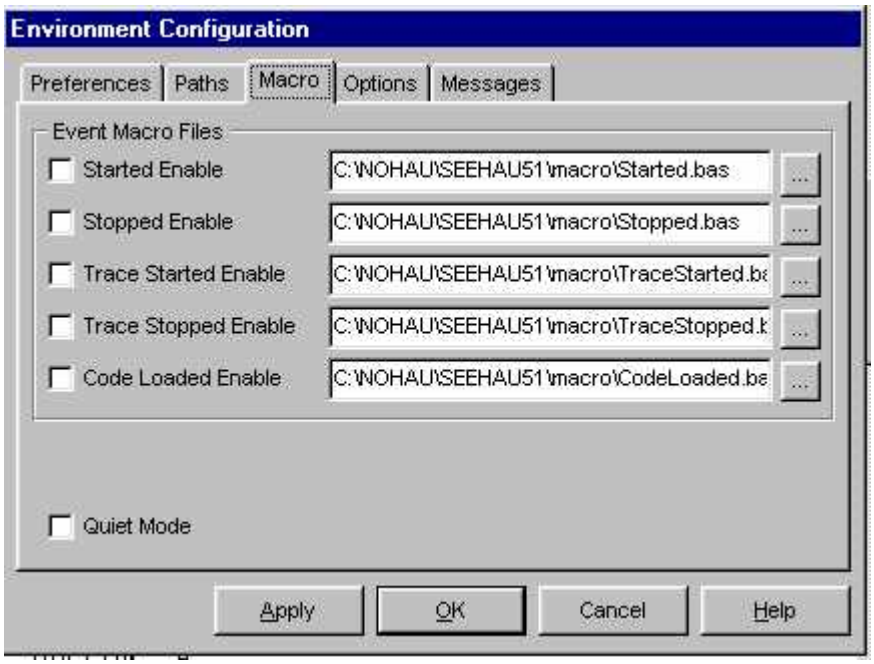
Nohau Emulator Environment and Configuration Setup

The following is an example configuration for the Noahu emulator when using the uPSD. The following screens show how to define and setup the memory map for the emulator as well as disable the watchdog, etc.

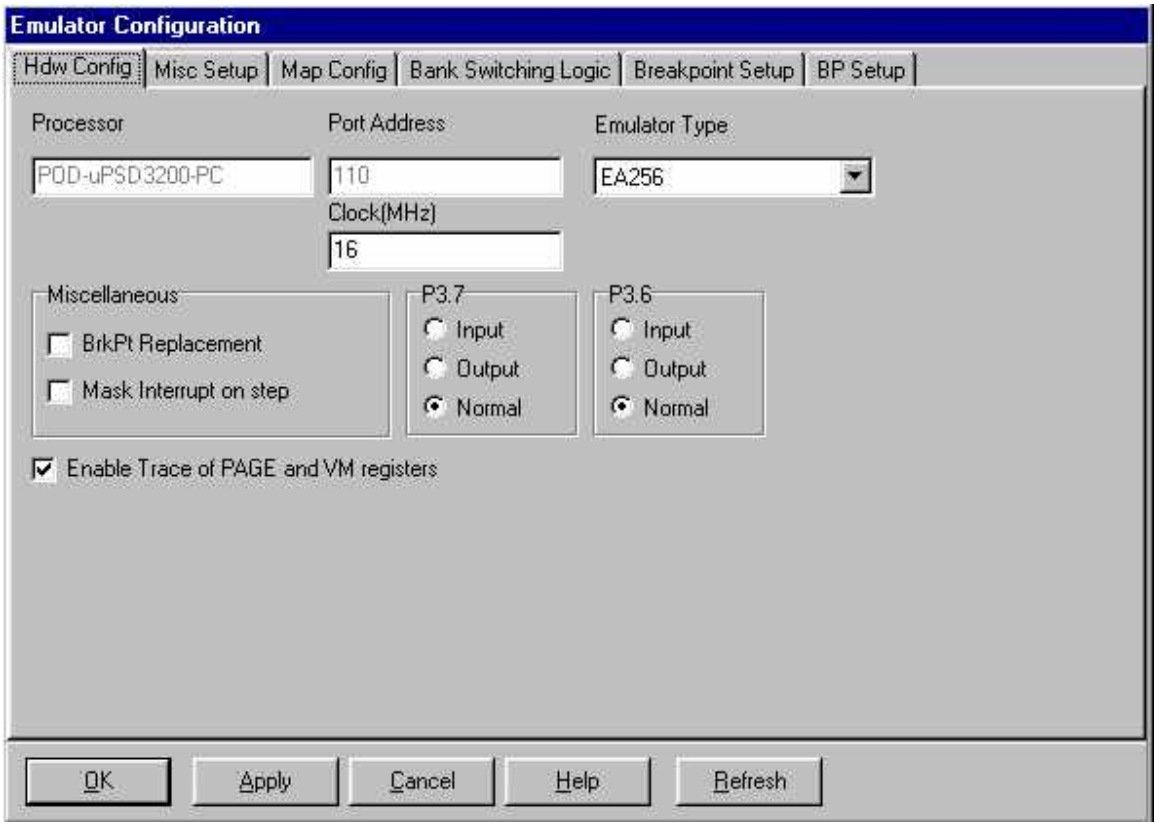
The emulator's Environment should be setup as follows:



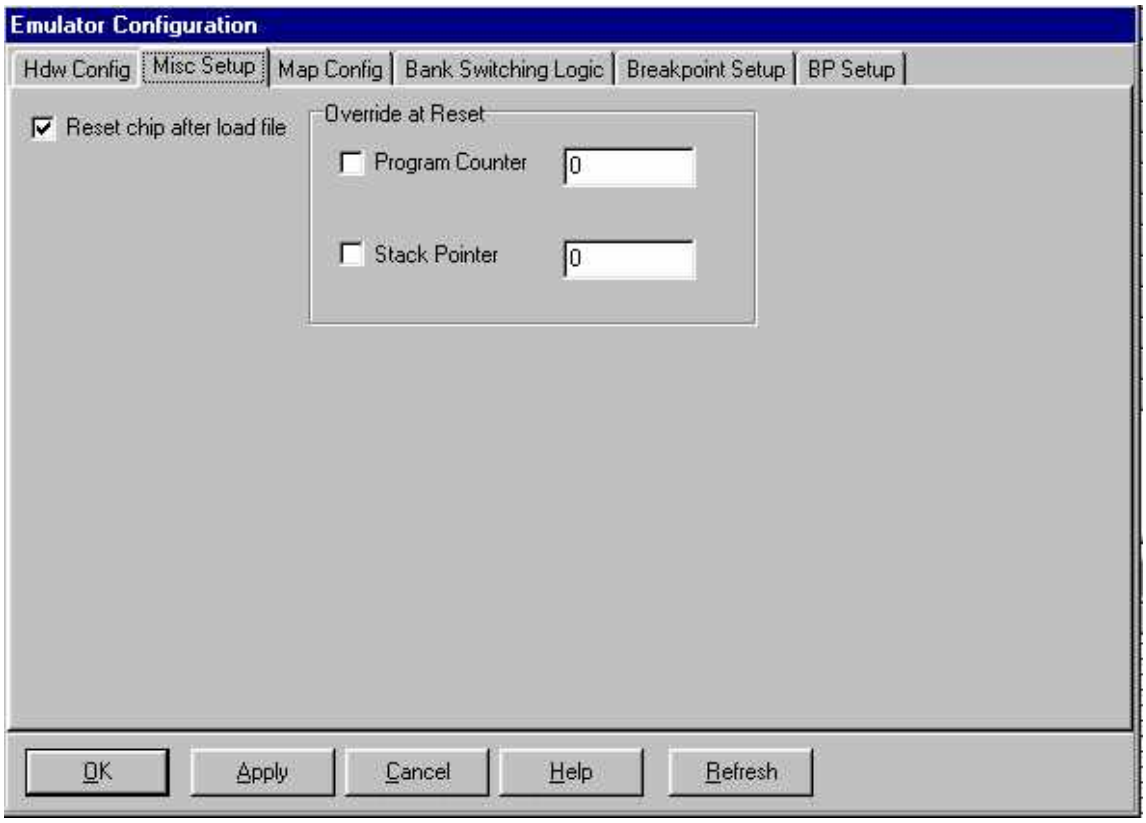


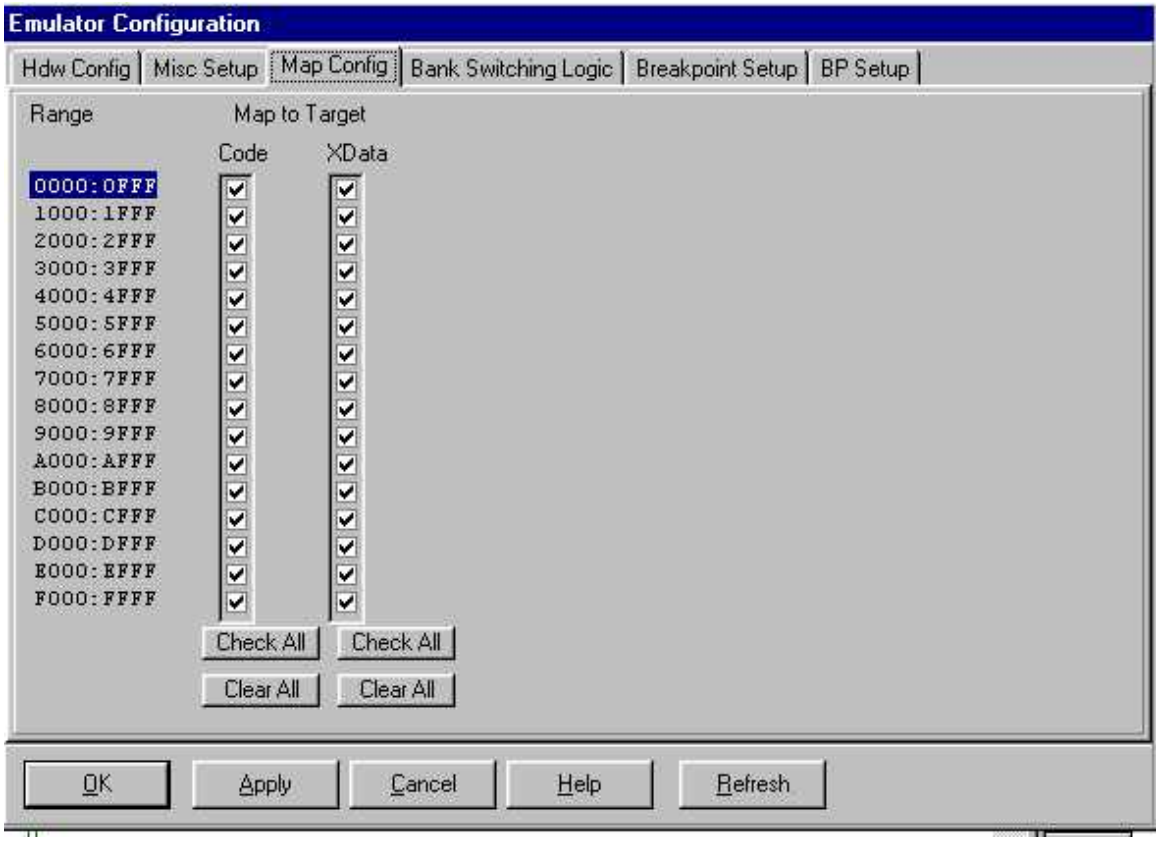

Environment Configuration
Preferences
Paths
Macro
Options
Messages
Event Macro Files
Started Enable
C:\NOHAU\SEEHAU51\macro\Started.bas
Stopped Enable
C:\NOHAU\SEEHAU51\macro\Stopped.bas
Trace Started Enable
C:\NOHAU\SEEHAU51\macro\TraceStarted.ba
Trace Stopped Enable
C:\NOHAU\SEEHAU51\macro\TraceStopped.b
Code Loaded Enable
C:\NOHAU\SEEHAU51\macro\CodeLoaded.ba
Quiet Mode
Apply
OK
Cancel
Help

The Emulator's Configuration should be setup up as:

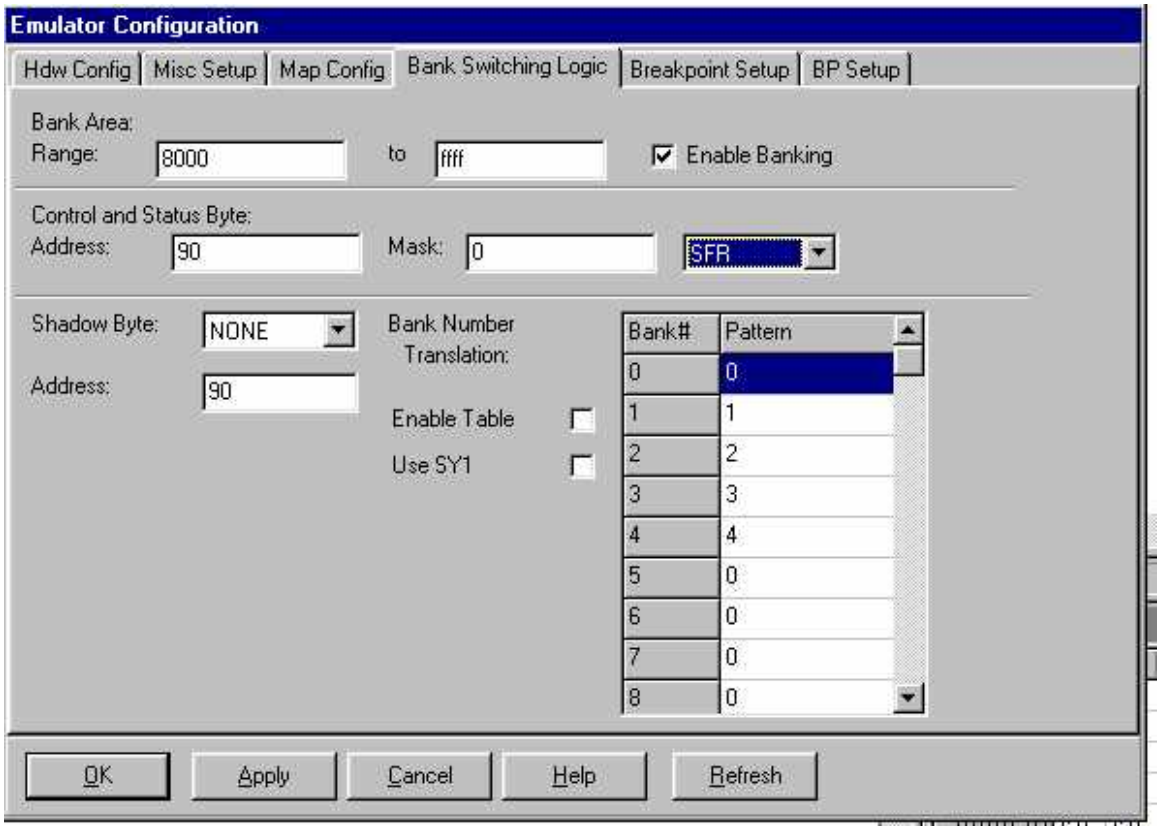


The default address range for the Emulator board is 110h – 11Fh. The emulator type is set to EA256, which means it supports bank switch for up to 256K memory. P3.6 and P3.7 are the two port pins that can be emulated by the pod and be used as the read and write signals, or can be used as normal port pins when set to Normal. Please see page 73 of the EMUL51-PC User Guide for more information on how to use them as read/write signals. The Enable Trace of Page and VM registers when checked, will allow the user to get those registers information displayed in the trace memory. The figure which is included under the Other Useful Features section, shows the Page Register content as well as the content of the VM register, i.e. if the code is in the Main Flash (MF), Boot Flash (BF), or SRAM (SR).

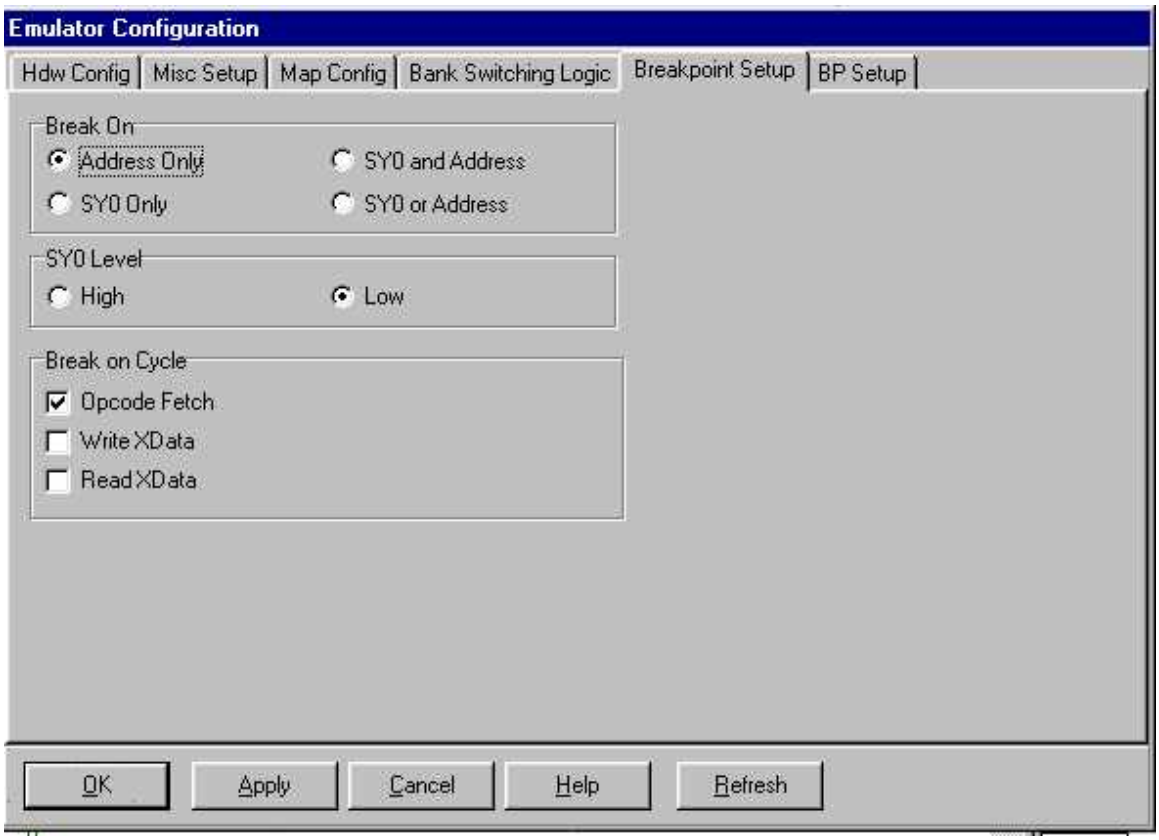

Emulator Configuration
Hdw Config
Misc Setup
Map Config
Bank Switching Logic
Breakpoint Setup
BP Setup
Reset chip after load file
Override at Reset
Program Counter
0
Stack Pointer
0
OK
Apply
Cancel
Help
Refresh



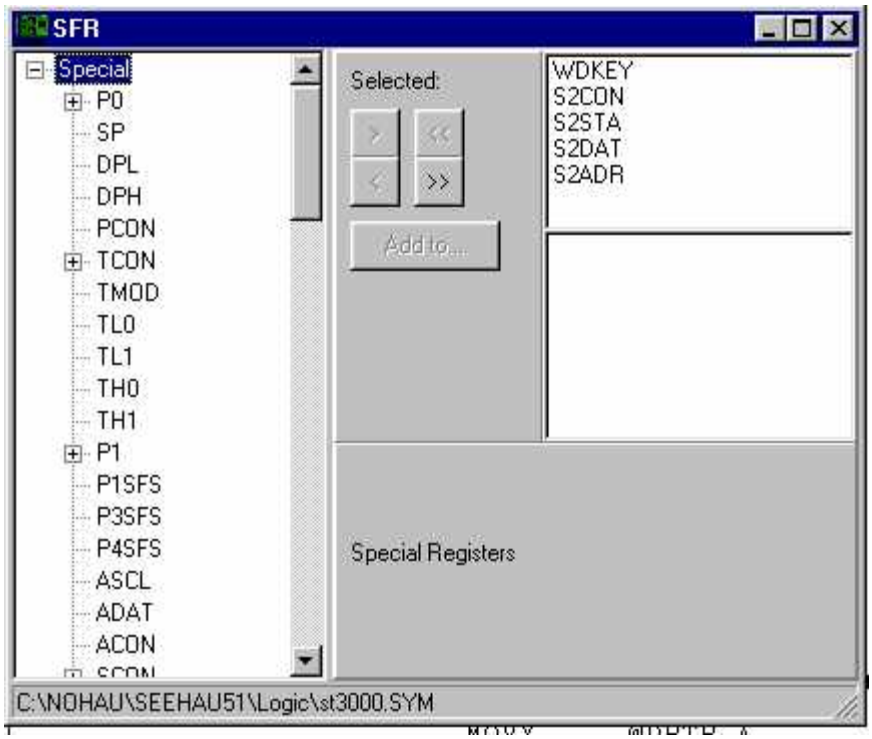
Click on the Check All button to map all the address spaces to the PSD, uPSD's PLD will perform the memory mapping for the MCU.

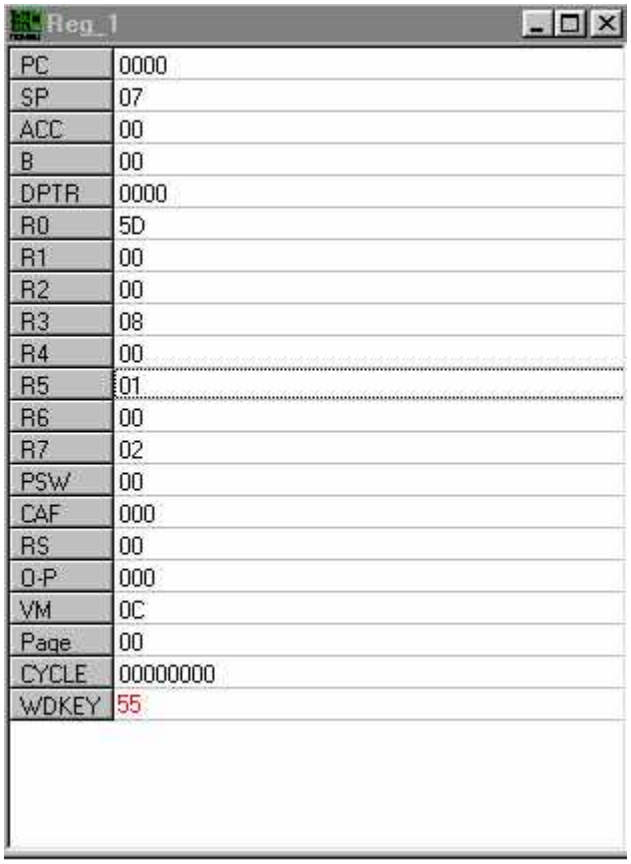


Please consult the The EMUL51-PC User Guide on how to setup the Bank Switch configuration. The BSW jumper block for the POD-ST3000 should be setup the same as the BM jumpers which is described in the EMUL51-PC User's Guide, page 25.

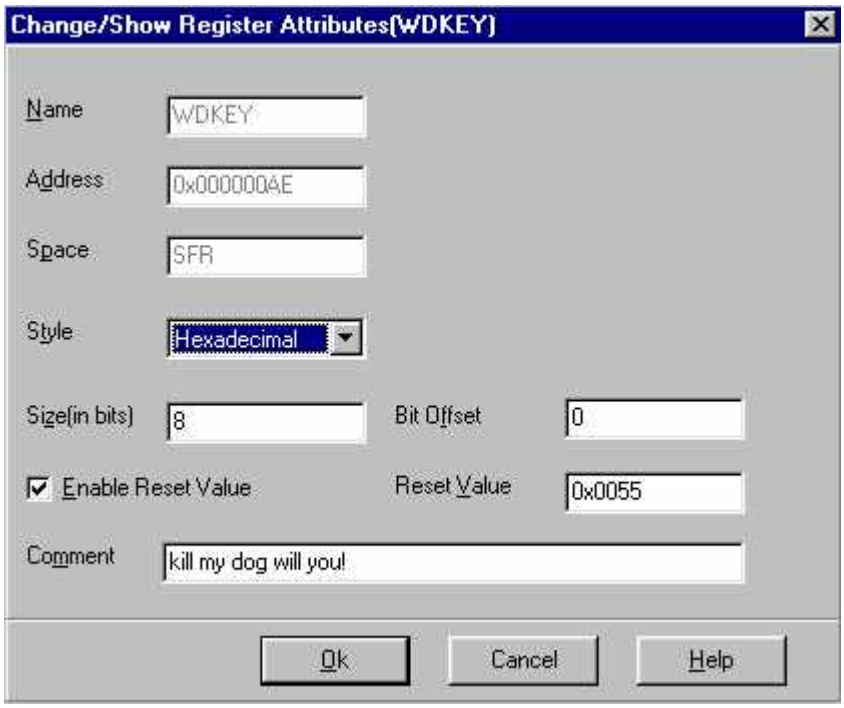


IMPORTANT:
Once the emulator's environment and configuration menus are setup, we also need to turn off the uPSD's Watchdog Timer since it will interfere with the emulator's operation. To turn off the Watchdog, place the cursor in the Reg_1 display window, then right click the mouse button, a menu with all the functions supported by the emulator for the register editing will appear, select SFR , then highlight WDKEY then click "Add to".


SFR
Special
P0
SP
DPL
DPH
PCON
TCON
TMOD
TL0
TL1
TH0
TH1
P1
P1SFS
P3SFS
P4SFS
ASCL
ADAT
ACON
Selected:
Add to...
WDKEY
S2CON
S2STA
S2DAT
S2ADR
Special Registers
C:\NOHAU\SEEHAU51\Logic\st3000.SYM


Reg_1
PC 0000
SP 07
ACC 00
B 00
DPTR 0000
R0 5D
R1 00
R2 00
R3 08
R4 00
R5 01
R6 00
R7 02
PSW 00
CAF 000
RS 00
O-P 000
VM 0C
Page 00
CYCLE 00000000
WDKEY 55

To permanently disable the Watchdog on reset, we need to setup the attribute for WDKEY register. To do so, click on the WDKEY register, then right click the mouse button, select "Attribute". Setup the WDKEY register attribute menu as follows:



The emulator is now ready to emulate. The emulator is operated by the Seehau user interface. Seehau is a high-level language user interface that allows you to do many useful tasks, such as load, run, single step programs based on C or Assembly languages.

The Getting Started Manual that comes with the Nohau Emulator Installation software CD for classic 8051 also applies to this emulator. The manual gives detailed information on how to use and setup certain features of the emulator, such as how to set break points, set triggers and view traces, modify and view memory contents including SFRs. Also please don't forget to use the HELP button, it has much helpful information on how to run the various emulator commands.

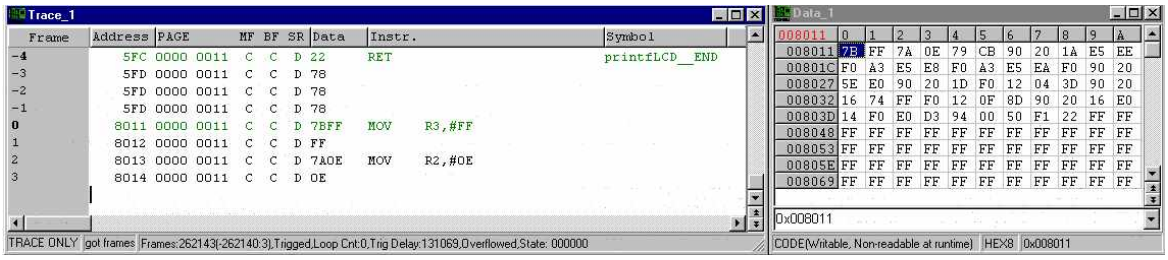
Emulator POD - Jumpers and Headers Setup

The jumpers and headers setup for the emulator pod is attached in the Appendix A. This document is taken from the Nohau's website at: http://www.nohau.com

Also from the same website, you can find the uPSD3200-POD's pin out and dimensions. It is included in the Appendix B of this document for your convince.

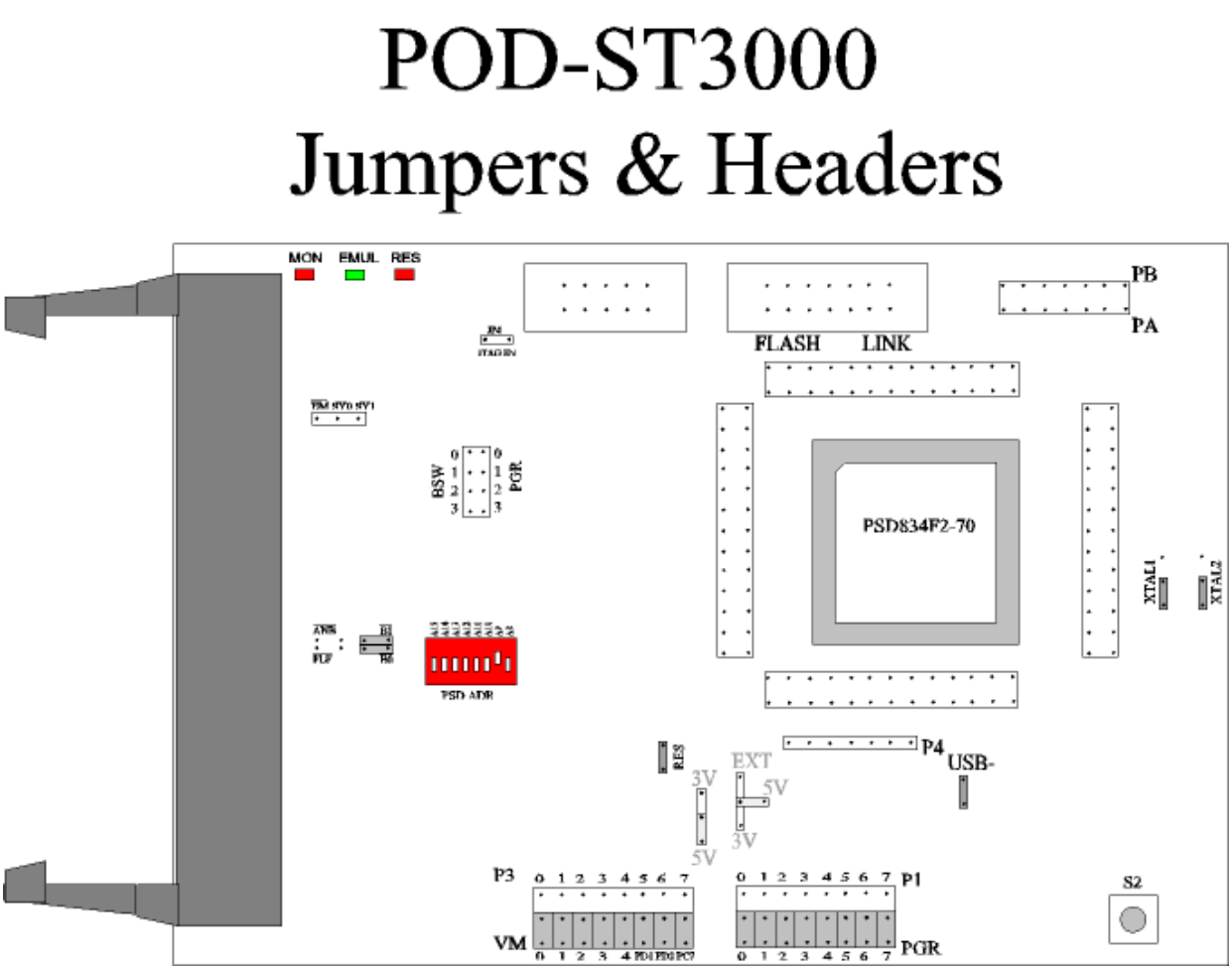
Other Useful Features

To edit the Source Code window, such as go to a certain address, or check where the break point is etc. can be accomplished by placing the cursor in the Source_1 window then right click the mouse button, all the command support by the emulator for source window editing will appear. The same is true when you place the cursor in the Trace window, then right click the mouse button for the commands that support all the trace window editing. The Trace window below shows the code is executed from the Main Flash memory page three, and the C under MF in the header, means page 3 of the Main Flash is designate as code space. The Data_1 window below shows the data content of page 3 at memory location 0x8011.



Appendix A

This document covers both the POD-ST3000 and the EA256-BSW emulator board. This system is designed to emulate the ST-Microelectronics µPSD3200 microcontroller.



POD board information and setup:

The pod has three LEDs, named MON, EMUL, RES:

- MON – is red, and means that the system is in monitor mode. In monitor mode, the processor is executing code that is internal to the emulator. This code is not user code, and is used to communicate with the host PC, to set up breakpoints, etc.

- EMUL – is green, and means that the system is in emulation mode. In emulation mode, the processor is executing the user's code from the emulation RAM or the targets PROM depending on the mapping in the emulator software.

- RES – is red, and means that the microcontroller RESET pin is LOW (active state).

This pod is equipped with a micro "DIP" switch that is labeled with **PSD ADR** you must set this jumper to match the address of the **CSIOP** that you have configured your project for, (sets the upper byte of the address).

The board has three jumpers for power and crystal connection to the emulation chip:

- PWR – is used to select power for the processor. Power should be supplied from the emulator and the jumper should be in either the **5v** or **3v** position depending on the type of microcontroller installed in the pod. With the power jumper in the left two pins, **EXT** position, then the power will come from the target system. The other jumper is for the PSD logic Vcc supply, both of these jumpers should be set to the same setting.

- XTAL – determines if the crystal or clock is taken from the target system or the on pod crystal**.** The jumpers should be in the top two positions for crystal to be supplied from the pod or the **INT** position. If the jumpers are in lower positions, **EXT** position, then the crystal/clock is supplied from the target system.

If you use an external clock, note that XTAL1 is an input and XTAL2 is left open.

All jumpers for PWR should be in the **5v** or **3v** position depending on the type of PSD device that you have installed and XTAL should be in the **INT, "internal"** position when no target is connected to the pod board.

RST connects the target RESET pin to the emulator. If the target system has a watchdog, it will probably interfere with the emulation and **RST** should then be removed.

The **USB** jumper must be installed in you are going to run the pod in **stand-alone** mode without a target otherwise the USB controller will keep resetting the micro, making the emulator unusable. Remove this jumper if you are installed on a target system that is designed to use the USB interface, otherwise leave the jumper installed.

Some other headers are labeled P4 (Port 4), PB (Port B), and PA (Port A) are available for monitoring via oscilloscope or other devices.

JP4 – is for future design operations to enable JTAG programming from the Emulator. This jumper should remain out.

BSW – header is for monitoring the bank bits for the FS0 … FS8 regions, or they can be connected to the PGR bits on the connector next to them if you wish to emulate the PSD portion of the memory using emulation RAM.

PGR – header for bits 0 through 3 of the PAGE register.

$\overline{\text{ANB}}$ / B1 and are to select between the bank switch bit 1 or the TRACE non-latched output signal when condition that is setup occurs.

$\overline{\text{FLF}}$ / B0 and are to select between the bank switch bit 1 or the TRACE latched output signal when condition that is setup occurs.

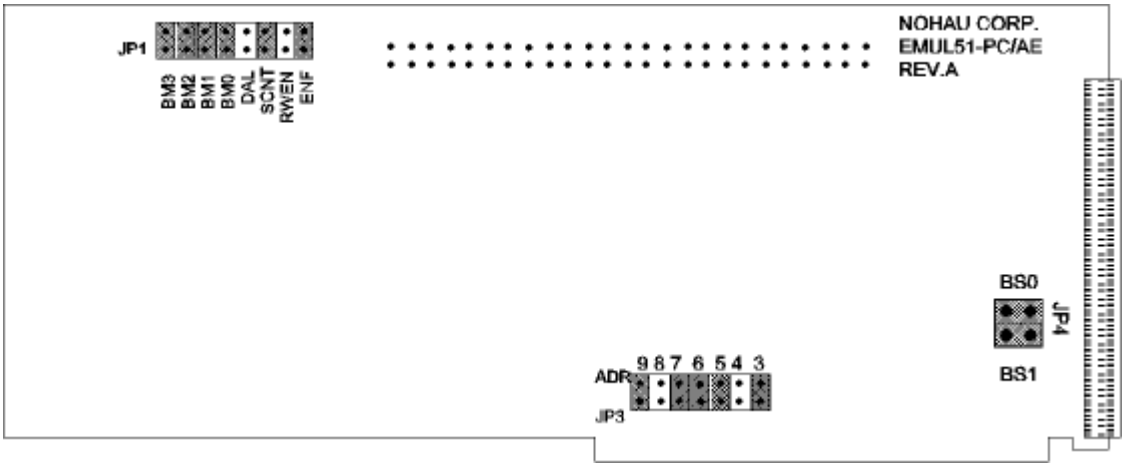
$\overline{\text{EM}}$ – this pin goes low when you start the emulator running code at full speed.

SY0 and **SY1** – these pins are general purpose inputs to the emulator and trace logic, and can be used to trigger on or break emulation.

Emulator board information and setup:

The emulation of the ST-Microelectronics µPSD3200 uses the EMUL51-PC/EA256-BSW emulator board. The figure below shows the correct settings for the emulator board, using default I/O address of 110, to correctly work with the POD-ST3000.

The emulator board must have COM 1.51 and be using software with a build date of 10/18/02 or newer to work correctly.



Appendix B

POD-uPSD3200 Diminsions