	AN815
![Micromint, Inc logo]	MicroBolt
JTAG debugging on the MicroBolt with the Nohau EMULARM	10/7/2005

Introduction:
This application notes demonstrates how to use Nohau's EMULARM to do in system JTAG debugging on the MicroBolt.

Background:
The MicroBolt contains a JTAG port for in system debugging and directly connects to the Nohau EMULARM (with 14 pin option - part number EMUL-ARM/ADP-14PIN). Nohau's debug format is directly supported by ImageCraft's ICCARM compiler, which also makes it a great choice for use with the MicroBolt.

The Nohau EMULARM JTAG debugger can be found here: http://www.nohau.com/emularm/emularm-pc.html

How it works:
As shown below, the EMULARM JTAG debugger plugs directly into the onboard JTAG debugging header of the MicroBolt. There are no special cable adapters needed given this direct connection, making it very easy to debug a MicroBolt while it's plugged into a system. The EMULARM is then plugged into the PC via a USB cable. This is all the setup that is needed since the EMULARM gets its power off from the MicroBolt.

MicroBolt with EMULARM plugged in:
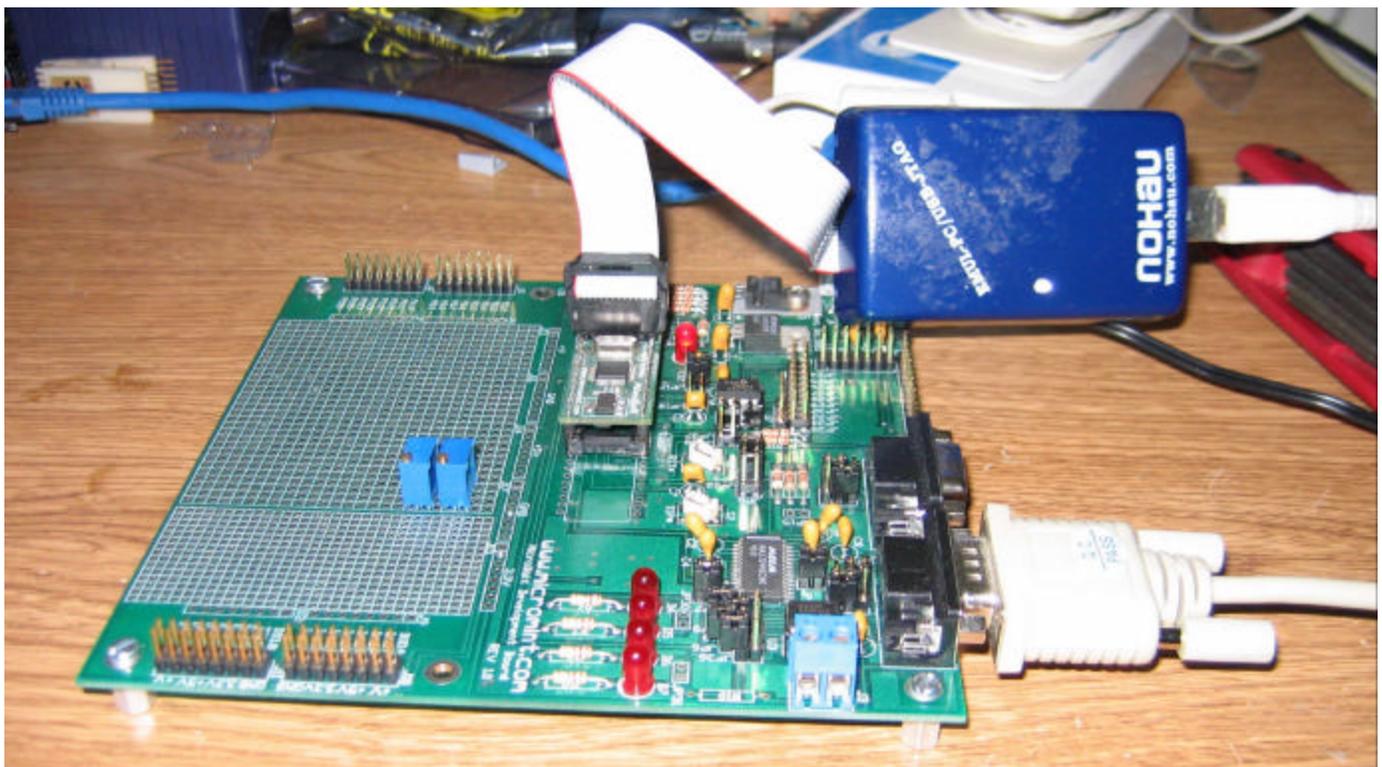
EMULARM Software:
The EMULARM comes with the Seehau user interface software. This software allows for in system debugging and an abundant amount of other features for debug. Besides the typical single stepping, breakpoints, and watch windows the user can view the MicroBolt's LPC2106 registers and modify them at will. The full list of Seehau features can be found on the Nohau website as noted above.

Seehau MicroBolt Configuration:
To start debugging with the EMULARM, the Seehau software must first be configured. The following steps and screenshots will show you how to set up Seehau to work with the MicroBolt.

Seehau Emulator Configuration – Hdw Config:
First, the hardware must be configured. If not already setup via the Seehau "Config" application, the following screenshot shows the recommended hardware configuration. The microprocessor is set to the Philips LPC2106, the "Reset method" to hardware, the clock to 60 MHz, the reset delay to 50 mS, and the rest are left as default settings.
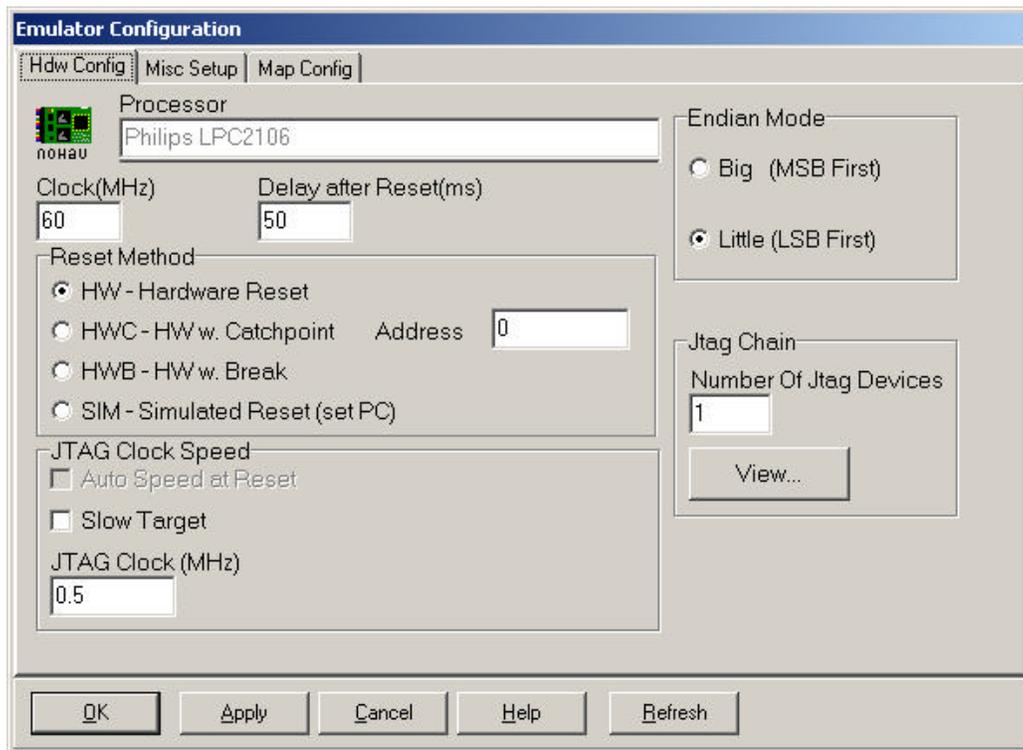
<u>Seehau Emulator Configuration – Misc Setup:</u>
Second, the misc. setup must be configured.   Configure the program counter and stack pointer to reset at 0, make sure "Remap after reset" is unchecked, check the "Reset chip after load", uncheck the "Disable flash programming" (very important step!), and set the "Target clock speed after reset" to 15 MHz (MicroBolt is actually 14.7456 MHz, but this gets it close).
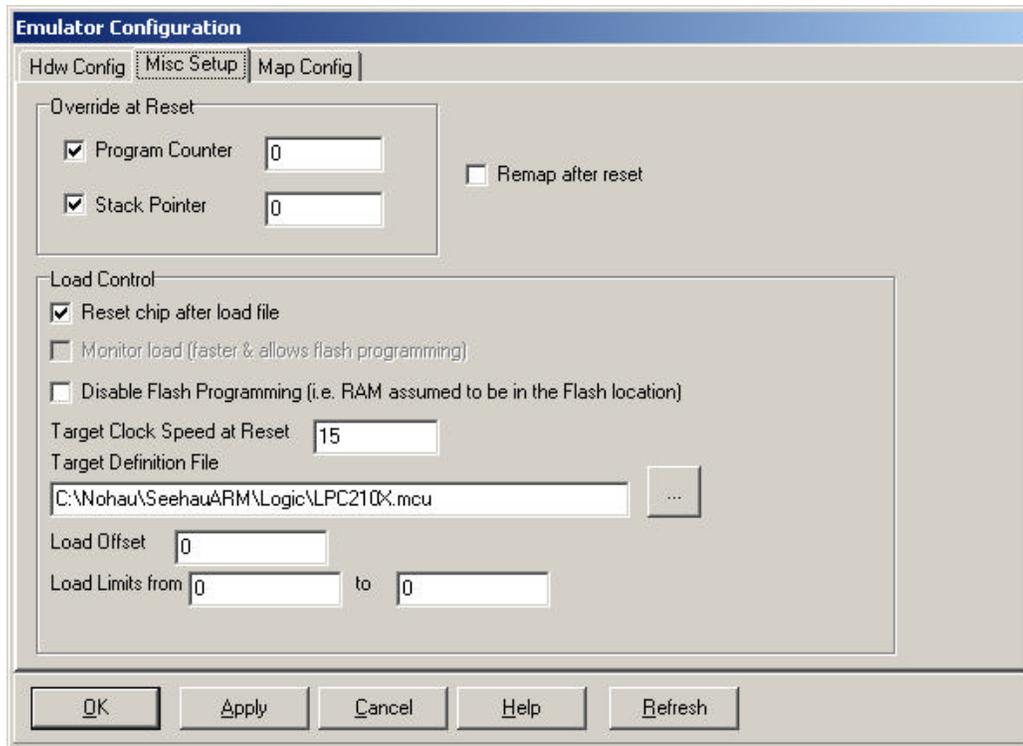
<u>Start Debugging with Seehau:</u>
You are now ready to start debugging with Seehau. The easiest thing to do is exit Seehau first. With the MicroBolt connected to the EMULARM, and your MicroBolt development board powered up, you can now restart Seehau with the new configuration changes.

<u>Seehau Reset Warning:</u>
If everything is hooked up and configured properly, the following warning will probably appear. Although Seehau states it's using simulated reset, it actually does a valid hardware reset of the MicroBolt. A hardware reset guarantees that the MicroBolt has been fully reset. Click OK and continue on.

Seehau - load code:

Now, you can load code into the MicroBolt. Go to any ImageCraft ICCARM demo project and load the .elf file as shown below. Select the "Verify After Load" to make sure your program was successfully programmed into the MicroBolt. Now you can run and stop your program, or set a breakpoint when the program is stopped. There are many other features and debug windows in Seehau. Information about these features can be found in the Seehau help files. A few of these features will be described below.
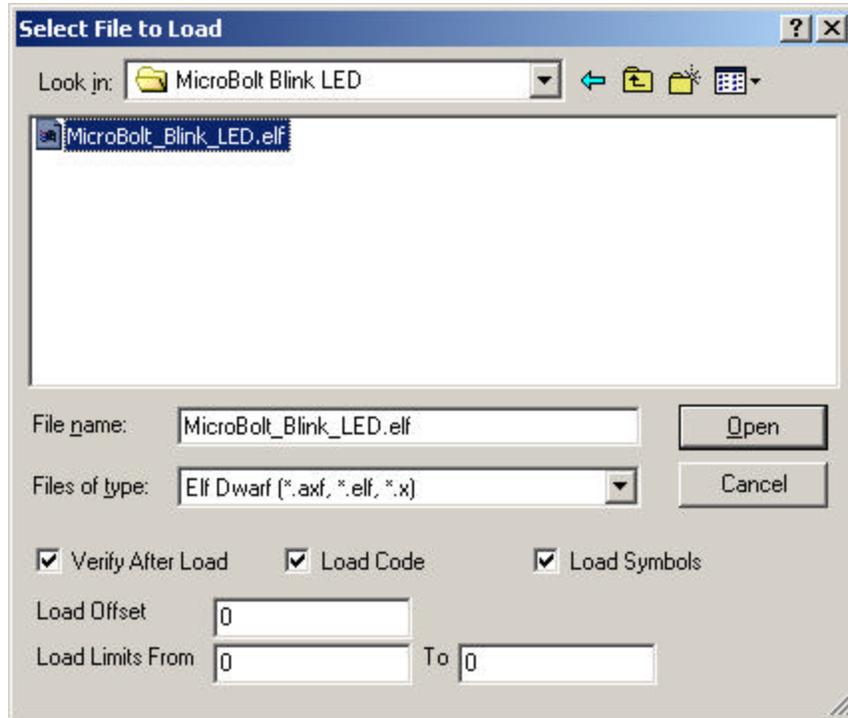
Seehau - Single Stepping:

With the EMULARM stopped, single stepping through the code can occur. The first thing that must be done is to select "Force Hardware Step" from the RUN menu. Now, various commands can be used in the "Run" menu.

Seehau - Using a Breakpoint:

With the EMULARM stopped, a hardware breakpoint can be set. This is done by pressing ALT+F2 or by right clicking and selecting "Breakpoints" and "Toggle Hardware Breakpoint". As a note, the MicroBolt's ARM based processing core only supports 2 hardware breakpoints.

Once the breakpoint has been set, run the code (F9 or the GO button). As shown below, when the breakpoint is reached in the code, the EMULARM stops and Seehau highlights the breakpoint in pink.
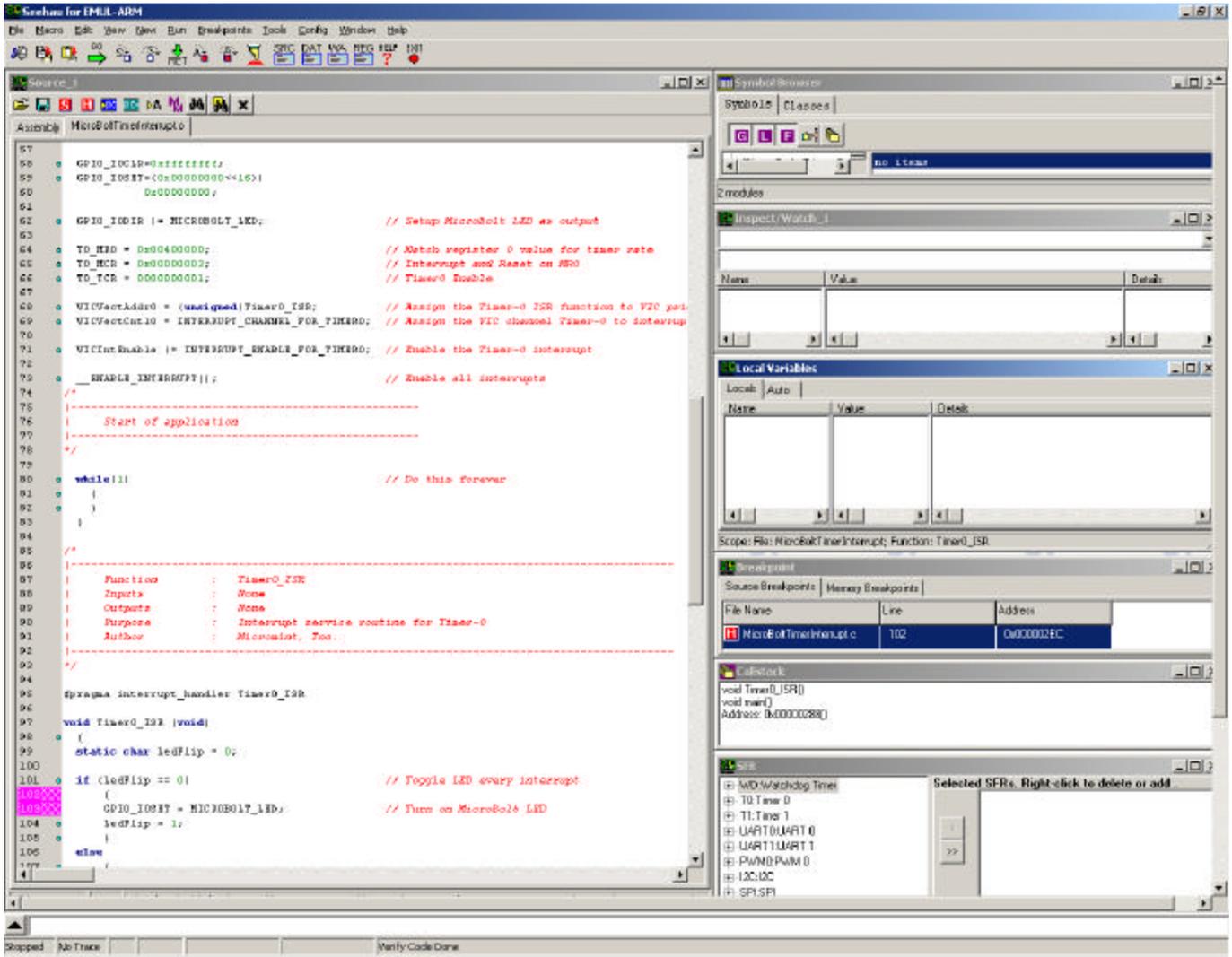
Conclusion:

The EMULARM with Seehau user interface is a powerful in system JTAG debugging solution. Given its direct support of the ImageCraft ICCARM debug format, it can be used with the various MicroBolt ImageCraft demo projects.