Connecting the Nohau EMUL-XAC3 Emulator to the PHYTEC phyCORE-XAC3 Rapid Development Kit.

Application Note # 105 *Version 2.5*

by Robert Boys
Nohau
Campbell, California (888) 886-6428 (408) 866-1820 www.nohau.com

May 22, 2000

Background

This application note is designed to help connect the Nohau EMUL-XAC3 emulator to the Phytec phyCORE-XAC3 evaluation board. It is assumed you have a working knowledge of the emulator. See the Nohau XA Getting Started Manual for more information. It is also possible to connect the XA-G3 emulator to this board with slightly different settings. Contact Nohau Technical Support at support@nohau.com or your local rep.

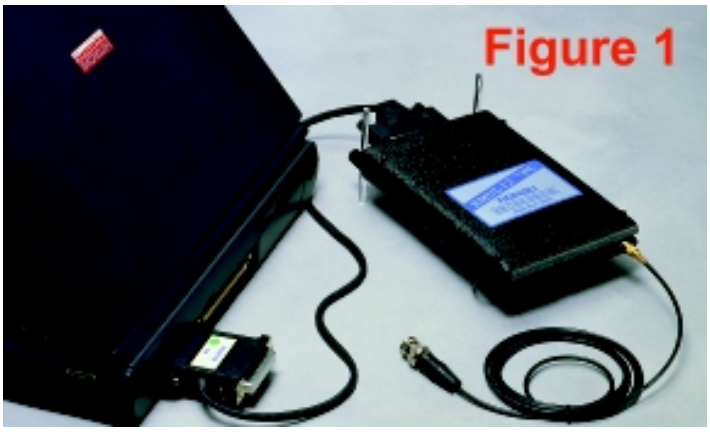
This application note will instruct you on the method of connecting the emulator to the Phytec board and get the entire system operating together. Instructions on operating the Phytec FLASH programming utility, FlashTools, via the emulator are given.

Running a simple demonstration program from both emulation RAM and the Phytec on-board FLASH are also given. Helpful hints to make this process easy are listed as appropriate as well as information regarding the XAC3 microcontroller or the emulator.

The Emulator

Nohau manufactures real-time in-circuit emulators offering complete support for the Philips P51XA microcontroller family. Controllers currently supported are the G37, C37, G49, SCC, H3 and H4. Note the G37 and C37 are commonly referred to as G3 and C3 respectively. The "7" refers to the OTP memory (one-time programmable). The "9" in G49 refers to the FLASH memory.

The emulator consists of an emulator board that is sandwiched with an optional trace board. Communication to your PC is via a LPT port or an ISA card. The user interface, Seehau, is a powerful debugging tool designed to run user code either stand-alone or connected to a user target system. The emulator is a handheld portable device and is shown in Figure 1.



The emulator can run in single-chip or expanded mode using the Philips bondout controller. See www.nohau.com for more information or contact your local Nohau representative.

The Phytec Rapid Development Kit

The Phytec development board as distributed by Philips consists as a small single board attached to a development board and this system constitutes the phyCORE-XAG3 Rapid Development Kit.

This system is pictured in Figure 2 showing the small single board computer with the P51XAC37 microcontroller installed in a PLCC 44 pin socket. The small single board computer is available with other processors such as the G49 and Philips 8051 derivatives such as the 87C591.

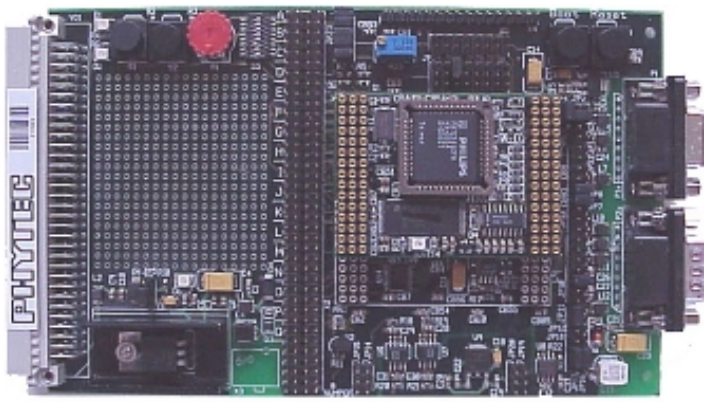


Figure 2

This system can be run stand-alone as pictured in Figure 1 or with the emulator connected as in Figure 4 to offer greater debugging, testing and confirmation power. This is accomplished by removing the C37 controller and plugging the emulator into the now empty PLCC socket and powering the emulator.

The user then has complete control and access to the target system resources. You can trigger on selective events such as a specified read or write of a certain value to an address or range of addresses. Data values can also be ranges.

You can contact Phytec at www.phytec.com. Their USA phone is (206) 780-9047 and Germany is +49 6131 9221-0.

Memory Organization of the phyCORE-XAC3

The Philips XA family uses separate code and data memory areas. This is called the Harvard Architecture. The emulator has a total of either 256K, 1M or 2M of emulation RAM depending on the model. The SCC/H3/H4 emulator has either 2 or 4M of memory. This memory may be used to replace memory that normally resides in the target. This emulation memory can be allocated to the code and/or data space in several configurations.

This note uses a 256K emulator with 128K mapped to the code space and 128K mapped to the data space. This is set in the Seehau software under the Config, emulator menu item and is the default value. The emulation memory wraps around once the 128K limit is reached. The data stored in location 0000 will be reflected in address 20000.

The emulator provides a mechanism to map the code or data memory areas either to this emulator RAM or to the memory devices on the target board. For example, you can use the RAM in the emulator to replace the FLASH in the target allowing easy and fast code loading without reprogramming the FLASH device. The target ROM or RAM can be mapped to the emulator with a granularity of 16 bytes.

FLASH Memory:

There is one FLASH device U3 which is a 29F160 512Kbyte by 16 bits device which represents 512K words of FLASH memory. The address range is from 0 to FFFFF hex. The upper 32K bytes is reserved for the Phytec FlashTools firmware. This area can be displayed with the emulator. The FLASH memory chip has the A0 pin connected to the CPU A1pin. Each increment by one in the address accesses one byte rather than one word.

Address 0000 will address 2 bytes at 0000 and 0001 in the memory map. Address 0002 similarly accesses 2 bytes at 0002 and 0003. The XA is a little endian machine which means the most significant byte of a word is stored at the higher address.

The emulator will display a word in 16 bit format as you expect to read it: i.e. 1234. In 8 bit format it will be displayed as it is stored in memory: i.e. 34 12. The emulator can run programs from either the FLASH devices or in the code emulation memory.

RAM Memory:

There are two RAM devices U4 and U5. They are 512K 8 bit devices for a total of 1 Mbytes or 512K words. Like the FLASH devices, the RAM A0 pins are connected to the CPU A1 pin. The address range is from 0 to FFFFF hex. The emulator can access this RAM or the emulation RAM for data storage.

The Bootstrap Loader

The "BOOT" button (S3) on the phyCORE-XAC3 development board enables the Phytec bootstrap loader function. The C3 chip does not have its own bootstrap loader: this is a Phytec implementation of their own and is described in their Hardware Manual.

In the Phytec directions, the bootstrap mode is entered by holding the RESET and BOOT buttons down together, releasing the RESET, then a few seconds later, the BOOT button.

With the emulator connected, the procedure is slightly different. The bootstrap mode is entered by ensuring the emulator is in the stop mode; it will say STOPPED in the bottom left hand corner. This is the emulator monitor mode. The Code and Data memory areas must be mapped to the target and is described later.

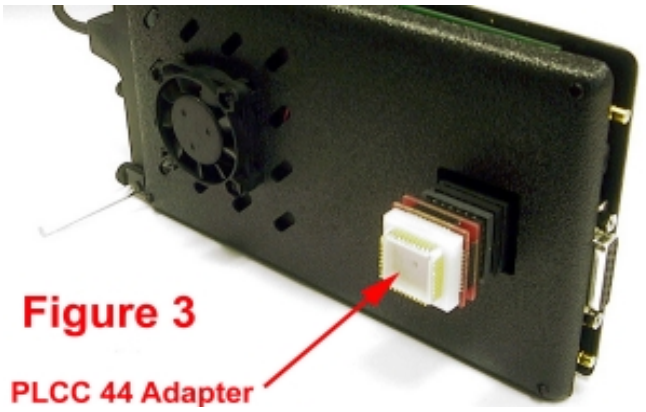
Hold the RESET and BOOT buttons down together and release the RESET. Still holding the BOOT down, press the RESET icon on the emulator and then press the GO icon. Release the BOOT button and the emulator will run the Phytec bootstrap mode firmware in the FLASH.

It is necessary to press the Phytec RESET button down in addition to the emulator RESET icon because the Phytec CPLD logic chip must be reset: the emulator does not send its RESET signal down to the board to avoid RESET line contentions. You can send the Phytec RESET up to the emulator by adding the RESET jumper JP18.

You still need to activate both RESET lines when JP18 is jumpered and the emulator is stopped (it is in *monitor mode*). This is in order to reset the Phytec CPLD device. But pressing the Phytec RESET button will reset both the board and the emulator and execution will continue as before with JP18 jumpered.

The Adapter

To connect to the Phytec board and standard PGA to 44 pin PLCC adapter is used. The Nohau part number is PGA 44-PLCC44. This is an industry standard adapter. Call Nohau Technical Support for ordering information. Figure 3 shows the bottom of the emulator with the adapter clearly shown by the arrow.



Powering the Phytec phyCORE-XAC3 Board

The Phytec board can be powered either by its own power supply via the 5 volt regulator U1 or by the EMUL-XA emulator. To power the Phytec board by the emulator, leave the jumper JP15 (POD PWR) installed on the emulator.

To power the Phytec board with an external power supply, take JP15 off to prevent damage to the emulator's bondout CPU. Remember to power the emulator always when power is supplied to the target with an external supply. You do not want the target to have power when the emulator does not.

If you use the emulator to provide the power, you do not need to worry about power sequencing. This method is recommended for this board. The emulator can provide perhaps 250 ma of power.

Do not confuse JP15 POD PWR with JP28 TARGET ON. Target ON is used to prevent leakage voltages from reaching the target during emulator powerup. See the Seehau Online Help about JP28.

Shut down Seehau and remove the power before connecting or disconnecting the target with the emulator. You can remove or add most jumpers on the emulator or the target while the system is powered but not running in emulation mode. Click on the RESET icon in Seehau to activate any new jumper settings. Reload the example program if you changed any address jumpers.

Connecting the Emulator to the phyCORE-XAC3 Board

It is a good idea to get the emulator working in stand-alone mode without being connected to the target system. This way you are sure your system is operational and you will be able to better isolate any problems potentially encountered.

See the Nohau XA Getting Started Manual for examples and hints on getting your system operational quickly.

The examples given in this note will be running the Phytec FlashTools utility via the emulator to check, erase and program the FLASH device. A simple example program will be loaded and run which will blink led D1 on the evaluation board at a rate determined by pressing the buttons S1 and S2.

Setting the Emulator Jumpers

The jumpers must be set as shown. These are the default values for the emulator in most cases except for the clock jumpers.

JP26	off	CODE
JP22	on	OVERLAY#
JP19	off	I/O PORT
JP20	on	EXT. MODE
JP27	off	8 bit
JP21	off	TARGET BW
JP16	5V	5V / 3V
JP14	on for EPC cable - off for ISA card	PC-PWR
JP28	on	TARGET ON
JP15	on	POD PWR
JP24	off	12 / 16 bit
JP25	off	12 bit
JP11	off	RXD0@J1
JP12	off	RXD0@J2
JP23	on TARGET	WAIT
JP18	on (or off)	RESET
JP3 - JP10	P2.x	
JP1 & JP2	See Clock Circuit information below.	

The PC-PWR jumper JP14 must be installed if you are using the EPC cable. OFF if you are using the ISA card. If JP14 is left on any power coming from the ISA card will cause a contention.

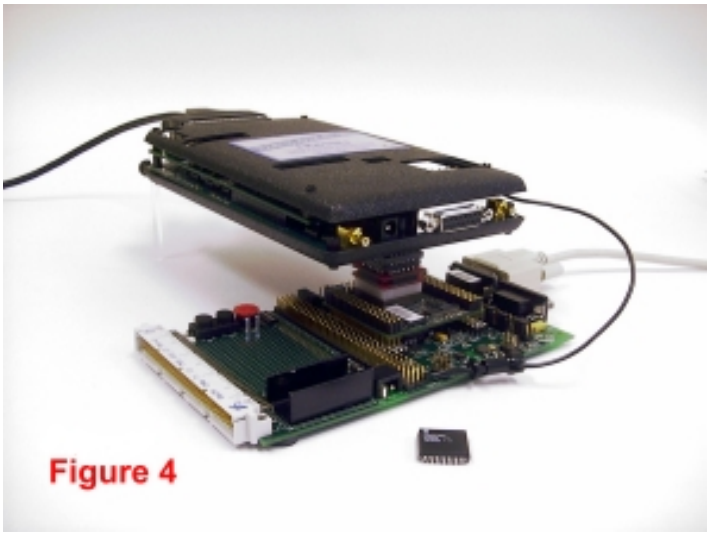


Figure 4

Clock Circuit: Emulator or Target ?

Connect the clock XTAL1 and XTAL2 (JP1 and JP2) to the TARGET position to use the phyCORE-XAC3 crystal. To select the emulator oscillator clock or the target crystal: there are two jumpers beside the crystal on the emulator to the target side. (JP1 and JP2). Problems starting up the emulator may happen because the crystal signal is not strong enough due to the extra lead length. In this case connect the two jumpers to the POD position and use the emulator oscillator.

If you use the emulator oscillator (POD position), you will need to send the clock signal down to the Phytec board for the Phytec CPLD logic chip. If this chip does not get this clock, the yellow READY light will come on indicating the WAIT line is active and the emulator will freeze.

It will do this because it is waiting for the WAIT signal to go inactive indicating the end of the memory wait time which will allow the memory cycle to finish.

The emulator does not send the clock signal to the target by default. Jumper all three connections of XTAL2 (JP2) together to select the emulator oscillator and send the clock to the target.

If you select the Target oscillator settings, the CPLD will get the clock signal and the three pins need not be jumpered.

The Phytec crystal is 32 MHz and the emulator is 30 MHz. This did not seem to matter. I tried a 25 MHz oscillator and the Phytec board still worked reliably.

Connecting the Emulator and Powering Up

Connect the emulator as shown in Figure 4. Power the emulator with the Phytec board connected. The green LED D6 beside the Phytec regulator chip will illuminate indicating power from the emulator. Start Seehau from your PC.

If the emulator does not start, check all connections and jumper settings. During the configuration process select POD-51XAG3-256/1E(EXT). Use the defaults for all other settings.

Try starting the emulator stand alone without being connected to the target or try switching the XTAL jumpers JP1 and JP2 to POD. Make sure the ground wire is connected to the right hand end connector of header X2. Leaving this ground wire off will not stop the emulator from working but connecting it will increase the system stability.

The photo in Figure 4 shows several black PGA extender adapters between the emulator and the PLCC plug. This was done for clarity. In practice all the extenders should be removed to improve high frequency operation. It is important to have all leads as short as possible, especially the clock signals, in adherence to good engineering practice.

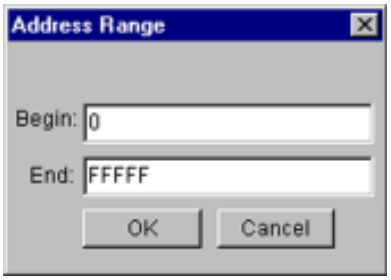
In the Seehau Source Window you should see something like the following if an example program is loaded in FLASH:

```
000200  D4073C00        FJMP 73C
000204  D4023A00        FJMP 23A
...                ...              ...
```

If the Phytec FLASH is blank, you will get FF BKPT in the source window. If good code comes up, the emulator is running: congratulations! - you have successfully got the system to work.

Some Helpful Notes and Reminders

1) The Target/pod WAIT jumper is important. It works on TARGET - but if you are using the emulator clock then the 3 pins of XTAL2 (JP2) need to be shorted together. See the notes under *Clock Circuit: Emulator or Target ?* You can set the WAIT jumper to POD, but no waitstates will be generated if they are needed.



2) Adapter - take out as many of the connector layers as possible on the adapter - make the path to the target as short as possible.
3) The Phytec RESET comes up to the emulator but RESET from the emulator does **not** go down to target. This means that if you perform an Emulator Reset you will have to reset the target with the RESET button to reset the Phytec CPLD.
4) If you have JP15 ON - POD PWR: supply 5 volt power to the emulator only and not the target. If you want to power both with their own separate power supplies, take JP15 off but **NEVER** power the target without the Emulator powered!! The bondout, like all other controllers, does not like power coming into its address and data pins.

   My choice: Power just the emulator - the green LED D6 on the Phytec board will light and you don't have to worry about the power sequencing. I tried both methods and it didn't seem to change anything.
5) Using the crystal on the target to supply CLK to the bondout is more unreliable than the emulator oscillator due to the extra lead length. The emulator oscillator provides a more stable signal. If you need to get the clock signal down to the target, you must jumper all three pins of JP2 (right beside the emulator oscillator U1).
6) Make sure the ground clip is connected to the target as shown in Figure 4.

Configuring the Memory Mapping

1) Select Config Emulator in the Seehau software:
2) Click on the memory map config tab - enter the code and data address ranges as displayed below:
3) Then click OK for both code and data windows.

   This maps the target memory into the emulator. When the emulator accesses a memory space, it will access the target's memory.
4) Note a small box appeared beside your entry: you can use this to temporarily park this setting. For instance, in order to run a program using the emulation RAM for code rather than the Phytec FLASH, deselect the Code box. The Data memory must be mapped to the target in order for the LEDS and the buttons to be accessed correctly.
5) Click OK to download your mapping information to the emulator. It will take a bit of time to download to the FPGAs. This maps all the XA address space. If you make the ranges smaller, it will take less time to map.

Save These Settings !

In the main Seehau screen: select Config - Save Settings - click on Yes to save your settings.

This way, if a problem occurs and Seehau exits: you will not have to re enter all those address values.

I selected the macro file startup.bas. You can create your own name.

Running an example program from emulation RAM:

Same set up as before but with no mapping of code memory space to the target. Please park this setting as described in **Configuring the Memory Mapping: 4)** above.

1) Select Config, Emulator and deselect the code memory box.

Note: The data memory must be mapped to the target or the LED and button signals will never reach the target. The demo program will look dead.

2) Click on OK - this will take a few seconds to remap the mapping memory arrays in the emulator.
3) Move JP23 to POD WAIT if not already.
4) Click on the emulator and target RESETs. Any order.
5) In Seehau: File - load code with the file Xademo3.abs

You can get this from www.nohau.com/bob/phytec/demo.zip

6) File - verify loaded code: Should have no errors.

   Note: Errors at this point are usually the result of trying to write to FLASH because the mapping is set incorrectly.
7) Press GO

Program will run: D1 should blink and buttons S1 and S2 will change the blink rate.

To prevent a crash, - hit the target reset button quick when you stop the emulator if you get the hour glass and you think it is not going to go away !

8) If you stop the emulator the C source code will appear in the source window. The trace window will display the last 131,000 instructions executed.

Getting FlashTools to work

You must select the code address mapping to access the target FLASH device. See Configuring the Memory Mapping.

1) Click on the emulator RESET icon. Press and hold the Target RESET and BOOT buttons together. Click the Emulator RESET again.
2) Release the target RESET button (keep holding BOOT) and then click on GO on the emulator. The GO should change to STOP and stay this way. If it returns to GO- something is wrong, do it over again. Check your settings.
3) Release the BOOT button and the Phytec board will be in its bootstrap mode.

Note: Only the Emulator green light should be on, the red and orange emulator LEDs should not be lit.

4) Start FlashTools ftpc.exe and select C3. Operate FlashTools as per the Phytec instructions.

5) REMEMBER - if the hourglass stays on for too long the emulator timeout may occur (especially when you stop the emulator program in some fashion) - press the Target RESET to prevent this if you sense this will happen !

3) You will be able to check, erase and program the FLASH memory on the Phytec board with the FlashTools utility.

Loading the Example Program into the Flash Memory

The emulator must map both code and data spaces to the target. See *Configuring the Memory Mapping* section for details.

1) Press and release the target RESET button. Click on the emulator RESET icon. The code shown on the previous page will show: FJMP 73C.

2) Press and hold the Phytec BOOT button and click on the emulator GO icon. The emulator should stay running.

3) Release the BOOT button and start the Phytec FlashTools utility. It will load normally.

4) Load the file Xademo3.hex from the demo.zip file.

5) Disconnect FlashTools via the Communications Setup tab in FlashTools.

6) The blinking program should run as before. You can start and stop execution. The trace memory can be viewed showing the last 131,071 instructions executed. Triggers can be set to many different qualifiers specified.

Using the EMUL-XA-G3 Emulator

The G3 emulator will work with the Phytec board but FlashTools will not due to some different C3 instructions that are used. It will run the demonstration program as described.

Using the EMUL-XA-G49 Emulator

Call Nohau Technical Support for information on using the EMUL-XAG49 emulator with the Phytec phyCORE-XAG49 board.

Conclusion

A simple demonstration has shown how easy it is to connect the EMUL-XA-C3 to a target hardware and run a target application. The Nohau C3 emulator provides an excellent debugging environment and works seamlessly with nearly all C compilers and assemblers. The Phytec boards provide an economical reference design that can be implemented in your final production run.

The Nohau EMUL-XA Emulator Description:

EMUL-XA Benefits

- **Philips P51XA-C3, G3, G49, S3, SCC, H3, H4.**
- **Full feature ICE. 16 to 32 MHz.**
- **Seehau GUI: Windows 95, 98, NT and 2000.**
- **Compact hand held design: goes anywhere.**
- **Optional 128K or 512K trace board.**
- **Supports all major C Compilers.**
- **Philips Bondout chip is used for emulation of internal and external modes.**
- **The Trace and Triggers are configured and accessible in real-time.**
- **No CPU cycle stealing.**
- **Designed and Made in the USA.**

Product Overview

The EMUL51XA-PC In-circuit emulator now supports the C3 microcontroller. It is a full feature, full speed emulator using Nohau's advanced design techniques and is designed and Made in the USA. The emulator consists of an emulation board and a compact trace board. The trace is optional and can be added later. The emulator is a hand held pod that is self contained. The emulator communicates with the PC through the parallel port. Seehau, the new Windows user interface provides exceptional HLL debugging with Windows 95/98/NT.

Single Chip and External Modes

Nohau supports the XA family for both external mode (ROMless) and internal mode (internal ROM) using pods containing a special Philips bondout chip for access to the internal address and data bus while leaving all ports intact and available for use. The emulator does not use any target system resources and not steal bondout cycles.

The emulator can operate stand-alone allowing debugging before your hardware is available. Adapters are available to connect to nearly any target board.

Breakpoints

Both software and hardware breakpoints are provided. Breakpoints do not execute the instruction they are set to and are unlimited in number.

Emulation Memory and Shadow RAM

The emulator has up to 1M of code memory and up to 1M of data memory. The granularity is 16 bytes offering mapping around nearly any peripheral address. Shadow RAM displays data writes in real-time. The data can be displayed in many numerical and graphical formats.

Trace Memory and Triggers:

Trace memory and the triggers are configurable and viewable without stealing CPU cycles. Full pipeline decoding ensures only executed instructions and data read/writes are stored and no false triggering occurs.

Triggers can be set on both addresses and data ranges. They control trace recording or cause the emulator to stop the target depending on the options set. Trace and Triggering can record all internal code and external code and data accesses: and in genuine real-time due.

The EMUL51XA-PC is supported everywhere with a world wide network of offices. Visit the Nohau web site or contact Nohau today for the name of your local Nohau representative.

nohau

Contact Information

Nohau **Tel: (800) 686-6428**

422 Peninsula Avenue **Tel: (650) 375-0409**

San Mateo CA 94401

sales@icetech.com

www.icetech.com