Connecting the Nohau EMUL51-PC Emulator to the PHYTEC phyCORE-89C51Rx2 Rapid Development Kit.

Background

This application note is designed to help connect the Nohau EMUL51-PC emulator to the PHYTEC phyCORE-89C51Rx2 Single Board Computer Module. It is assumed you have a working knowledge of the emulator. See the EMUL-51 Getting Started Manual for more information.

The Emulator

Nohau manufactures real-time in-circuit emulators offering complete support for the Philips 89C51RX2 microcontroller family. Controllers currently supported are the P89C51RB2,C2 and D2. The emulator consists of a High Speed Parallel (HSP) box, that contains an emulator board and optional trace board. The P89C51RD2 pod connects to the HSP box with a ribbon cable. Communication to the PC is via the 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 stand-alone device and is shown in Figure 1. The emulator can run in single-chip or expanded mode using the Philips "hooks" mode emulation technique. See www.nohau.com for more information or contact your local Nohau representative.

Figure 1

The PHYTEC Rapid Development Kit

The PHYTEC Rapid Development Kit as distributed by Philips consists of a small phyCORE-89C51Rx2 single board computer module attached to a development board. This system is pictured in Figure 2 showing the small single board computer with the P89C51RD2 microcontroller installed in a PLCC 44 pin socket. This system can be run stand-alone as pictured in Fig.2 or with the emulator connected, as shown in Fig. 4 to offer greater debugging and testing This is accomplished by removing the 89C51Rx2 controller and plugging the emulator into the PLCC socket. The user then has complete control and access to the target system resources. You can contact PHYTEC at www.phytec.com. USA phone is (206) 780-9047 and Germany is +49 6131 9221-0.
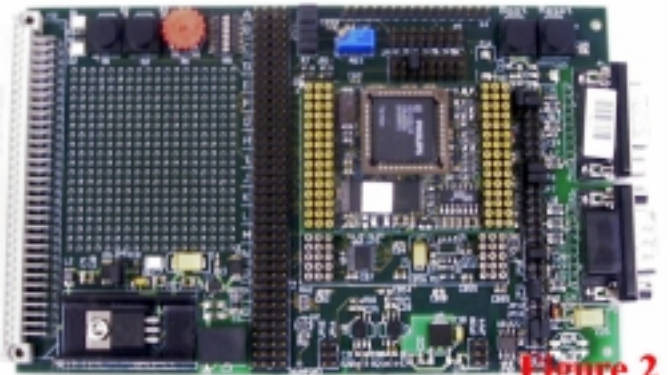

Figure 2

The Adapter

To connect to the PHYTEC board a standard PGA to 44 pin PLCC adapter is used. The Nohau part number is PGA44-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 Pin 1 Notch clearly shown by the arrow.
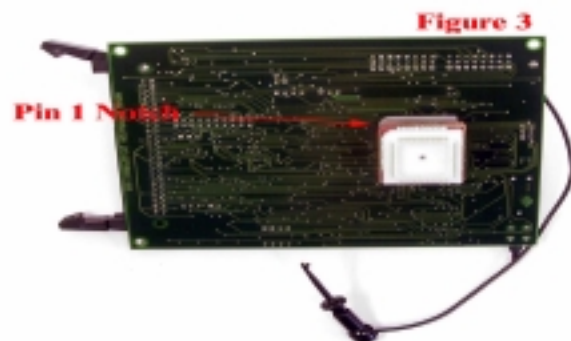

Figure 3

Pin 1 Notch

Memory Organization of the phyCORE-89C51Rx2

The Philips 8051 family uses separate code and data memory areas. This is called the Harvard Architecture. The emulator has a total of either 256 or 768K of emulation RAM depending on the model. 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 64K mapped to the code space and 64K mapped to the data space. This is set in the Seehau software under the Config, emulator menu item and is the default configuration. The 89C51Rx2 has a 64K address space for code and 64K for data. The additional memory is available for bank switched applications.

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 4K bytes.

FLASH Memory:

The phyCORE module has only the internal 64K flash of the 8089C51Rx2. The EMUL-51 does not support programming of the on chip FLASH. For debugging of programs residing in on chip memory put the JP4 jumper labeled M1 on and map all code memory to the emulator. Since the emulation memory imposes no wait states this will run at the speed as in the on chip FLASH memory. It also provides an easy method to download programs.

RAM Memory:

There is one RAM device U6. The address range is from 0 to 7FFF hex. The emulator can access this RAM or the emulation RAM for data storage.

Powering the PHYTEC phyCORE-89C51Rx2 Board

The PHYTEC board needs be powered by its own power supply via the 5 volt regulator U1 on the development board. Remember to power the emulator first and the target second. You do not want the target to have power when the emulator does not.

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-89C51Rx2 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 51 Getting Started Manual for examples and hints on getting your system operational quickly. A simple example program will be loaded and run which will blink LED D1 on the valuation board at a rate of once per second.

Setting the Emulator Jumpers

The jumpers must be set as shown. The default settings are marked *(DEFAULT)*, and the settings specifically for the phy-CORE-89C51Rx2 board *are marked (phyCORE)*.

Mode JP4

These jumpers labeled M1 and M2 which are used to select single chip (internal ROM), or expanded mode (external CODE and XData), and if P0,P3.6 and P3.7 are general purpose I/O,.

Single Chip

(DEFAULT)	M1	on	CODE
(DEFAULT)	M2	on	XData

Expanded Mode

(phyCORE)	M1	off	CODE
(phyCORE)	M2	off	XData

Internal Code, XData Mode

	M1	on	CODE
	M2	off	XData

Processor Power

(DEFAULT)	XJP2	INT	From pod
	XPJ2	EXT	From target

Reset

(DEFAULT)	JP5	on	RESET from target

Clock

(DEFAULT)	JP6 & 7	I	Internal clock
		E	Target clock

See Clock Circuit information below for more details.

Clock Circuit: Emulator or Target ?

There are two jumpers beside the crystal on the emulator (JP6 and JP7). Connect the clock XTAL1 and XTAL2 (JP6 and JP6) to the E (external) position to use the phyCORE crystal. To select the emulator oscillator clock connect JP6 and JP7 to the I (internal) position. 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 I position and use the emulator oscillator.

Connecting the Emulator and Powering Up

Connect the emulator as shown in Figure 4. Power the emulator first, then the target. Start Seehau from your PC. If the emulator does not start, check all connections and jumper settings. During the configuration process select 51Rx2-P. 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.

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.

Some Helpful Notes and Reminders

1) The M2 jumper is important. It must be off to access XData

on the target. The LED D1 will not blink if this jumper is set wrong.

2) The XData must be mapped to Target for the LED D1 to work.

3) Adapter - take out as many of the connector layers as possible on the adapter - make the path to the target as short as possible.

4) 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.

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.

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 - the memory config tab is displayed below:

Figure 5

3) Then click the All button under XData-Target.

This maps the target XData memory into the emulator. When the emulator accesses this memory space, it will access the target's memory.

Figure 4
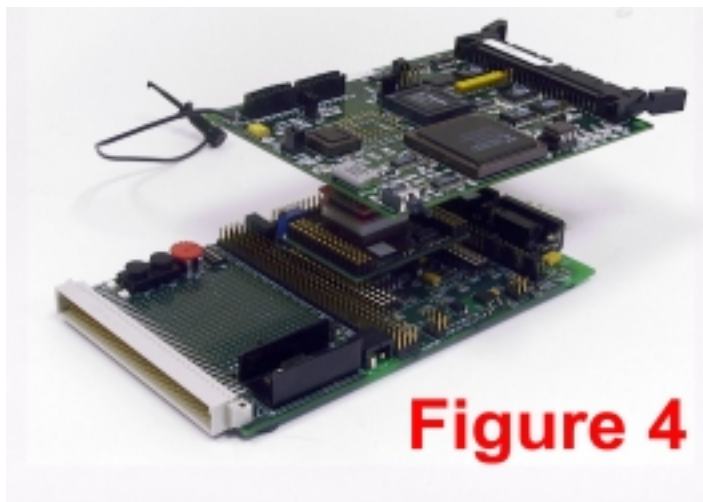
4) Note with this setting your program will run using the emulation RAM for code. The XData memory must be mapped to the target in order for the LEDs to work.

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 EMUL-51 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 to save the settings in but you can select any or name your own and call it any way you want using Seehau's settings.

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 select all for code memory.

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) Click on the emulator reset Icon and the target RESET button in any order.

4) In Seehau: File - load code. Browse to the directory \examples\phyRx2\keil\blinky and load the file blinky, as shown in Fig. 6.

Figure 6
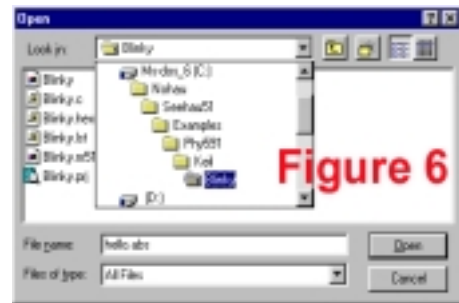
5) Next select File - verify loaded code: Should have no errors.

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

6) Press GO

Program will run: D1 should blink at a rate of once per second.

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 !

7) 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.

Conclusion

A simple demonstration has shown how easy it is to connect the EMUL-80C51Rx2 to a target hardware and run a target application.