VS Vision Systems GmbH
Industrielle Bildverarbeitung
Aspelohe 27A; 22848 Norderstedt
Tel : +49 (40) 528 401-0
Fax : +49 (40) 528 401-99
eMail: Multi@visionsystems.hh.uunet.de

Configure serial Ports with LINUX

There are no special drivers for the VScom-boards. This is not a problem, since the serial ports are supported by the kernel. Only the actual settings have to be configured. This is done with **Setserial**. The settings should be placed in **/sbin/init.d/serial**, to execute them at every system start. An example might look like this:

${SETSERIAL} cua0 port 0x3f8 irq 4 uart 16550A ^fourport
${SETSERIAL} cua1 port 0x2f8 irq 3 uart 16550A ^fourport
${SETSERIAL} cua2 port 0x100 irq 5 uart 16550A ^fourport
${SETSERIAL} cua3 port 0x108 irq 5 uart 16550A ^fourport
${SETSERIAL} cua4 port 0x110 irq 5 uart 16550A ^fourport
${SETSERIAL} cua5 port 0x118 irq 5 uart 16550A ^fourport

The ports cua0 and cua1 already exist, typical on the Mainboard. Ports cua2 through cua5 belong to a VScom 400, configured for base address 0x100 and IRQ 5. With a VScom 800 one will place settings up to cua9. Right now all ports work OK with these settings. Additional calls of **Setserial** may integrate more ports into the system.

There is a problem with **VScom PCI-Series** here. Via PCI-Bus (Plug&Play) the board gets its resources dynamically. So the settings aren't controllable directly. But Linux is sufficiently informative also here. In the file **/proc/pci** this might be found for a VScom 400 PCI:

Bus 0, device 11, function 0:
Serial controller: PLX Unknown device (rev 1).
Vendor id=10b5. Device id=1077.
Medium devsel. Fast back-to-back capable. *IRQ 10*.
Non-prefetchable 32 bit memory at 0xdf000000.
I/O at 0xb000.
I/O at *0xa800*.
I/O at *0xa400*.

Here the IRQ 10 and the base addresses are important.

0xb000: PLX base address (PCI Interface, don't care here)
0xa800: UART block address (the ports start here)
0xa400: Interrupt Vector Address, more information later.

The IRQ 10 is a parameter at **Setserial**, the ports are located consecutively in steps of **8** beginning at the base address. The resources are assigned dynamically with every start of the PC; but the only change, if something else is changed in the configuration of the computer. So one can rely on these values to be stable.

Additional the VScom PCI board is set to high transmission speed automatically, this is eight times normal. This would always cause a higher baud rate than configured by a program. To correct this, **Setserial** needs an extra parameter, this is **baud_base 921600**. The driver is configured to calculate all speeds based on a maximum transmission rate of 921.6kbps.

IRQ-Vector aka. fourport option:

The boards of series VScom 400 and VScom 800 offer an extra IRQ-Vector (or IRQ-Status) register. With this register the option of IRQ-sharing may be used more efficiently.

Technical: the register is implemented as a LS240 line driver, the inputs connected to the IRQ-outputs of the UARTs. By reading the values are placed to the data bus. Now bit 0 corresponds with the first UART (COMA, Port 1), bit 1 with the second up to bit 3 (or bit 7 on the VScom 800). In idle state a **"1"** is read here, in the active state a **"0"**.

This register may be used to reduce the service time of an interrupt significantly. Activated is this via the **set_multiport** option.

${SETSERIAL} cua2 set_multiport port1 0xa400 mask1 0xf match1 0xf

With a VScom 800 mask1 is 0xff, also match1, to control all 8 ports.

VS Vision Systems GmbH
Industrielle Bildverarbeitung
Aspelohe 27A; 22848 Norderstedt
Tel : +49 (40) 528 401-0
Fax : +49 (40) 528 401-99
eMail: Multi@visionsystems.hh.uunet.de

Compatibility and higher speed:

Especially older programs do not allow to select all speeds (namely the high ones). For these purposes there are the options spd_hi and spd_vhi. When a port is configured with this setting, the driver replaces the baud rate 38400 with the speed of 57.6kbps (or 115.2kbps). This setting is placed at the end of the configuration command line.

Setting the Handshake:

Stty – set terminal characteristics

If your modem supports hardware-handshake, you should ensure, this is active. Most surprising only few communication programs try to set this; so you must do this manually. This is best done also in the serial-Script, by calling **stty** this way:

$ stty crtscts < /dev/cua1

This is a complete Script for the VScom 400 PCI from the upper example:

${SETSERIAL} cua0 port 0x3f8 irq 4 uart 16550A ^fourport spd_hi
${SETSERIAL} cua1 port 0x2f8 irq 3 uart 16550A ^fourport spd_hi
${SETSERIAL} cua2 port 0xa800 irq 10 uart 16550A baud_base 921600 ^fourport spd_hi
${SETSERIAL} cua3 port 0xa808 irq 10 uart 16550A baud_base 921600 ^fourport spd_hi
${SETSERIAL} cua4 port 0xa810 irq 10 uart 16550A baud_base 921600 ^fourport spd_hi
${SETSERIAL} cua5 port 0xa818 irq 10 uart 16550A baud_base 921600 ^fourport spd_hi
${SETSERIAL} cua2 set_multiport port1 0xa400 mask1 0xf match1 0xf
$ stty crtscts 38400 </dev/cua0
$ stty crtscts 38400 </dev/cua1
$ stty crtscts 38400 </dev/cua2
$ stty crtscts 38400 </dev/cua3
$ stty crtscts 38400 </dev/cua4
$ stty crtscts 38400 </dev/cua5

With a VScom Turbo- or Pro-series card please insert the configured addresses.

The boards of the VScom Turbo – series allow different maximum speeds. In the PRO and PCI series the factors **1** and **8** are possible. Depending on configuration possible values for **Baud_base** are:

	Factor	**Baud Base**
All boards	1	115,200 kbps
VScom Turbo, VScom 200 PRO	2	230,400 kbps
VScom Turbo, VScom 200 PRO, VScom 100 PRO	4	460,800 kbps
VScom PRO, except VScom 200 PRO, VScom 100 PRO	8	921,600 kbps
VScom 100 PRO	12	1,382400 Mbps

The register IRQ-Vector is available on all boards VScom 400 and VScom 800 of the PRO- and PCI-series; on the boards VScom 400 Turbo and VScom 800 Turbo only in the so called Unix-Mode.

The upper settings – namely for the VScom 400 PCI – have been tested an transmitted by Mr. Stephan Isringhausen-Bley of the Ruhr Universität Bochum. Here again we want to express our many thanks for this.