



**VS Vision Systems GmbH**

Industrielle Bildverarbeitung

Aspeloh 27A; 22848 Norderstedt

Tel : +49 (40) 528 401-0

Fax : +49 (40) 528 401-99

eMail: Multi@visionsystems.hh.uunet.de

---

## Serielle Ports unter LINUX einrichten

Für die VScom-Karten werden keine speziellen Linux-Treiber angeboten. Dies ist kein Hindernis, da die seriellen Schnittstellen bereits vom Kernel unterstützt werden. Die Einstellungen müssen lediglich konfiguriert werden. Diese Konfiguration geschieht mittels **Setserial**. Die Eintragungen sollten in **/sbin/init.d/serial** vorgenommen werden, damit sie bei jedem Start ausgeführt werden. Ein Beispiel könnte so aussehen:

```
#{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
```

Die Ports cua0 und cua1 sind die schon vorhandenen Schnittstellen. Die Ports cua2 bis cua5 gehören zu einer VScom 400, eingestellt auf Basisadresse 0x100 und IRQ 5. Für eine VScom 800 wird entsprechend bis cua9 eingetragen. Mit diesen Einstellungen arbeiten bereits alle Schnittstellen korrekt. Mit zusätzlichen Aufrufen lassen sich auch weitere Schnittstellen in das System integrieren.

Bei der **VScom PCI-Serie** ergibt sich hier ein Problem. Durch den PCI-Bus (Plug&Play) erhält die Karte ihre Ressourcen dynamisch zugeteilt. Damit ist die Einstellung nicht direkt zu kontrollieren. Allerdings ist Linux hier ebenfalls ausreichende informativ. In der Datei **/proc/pci** findet sich z.B. für eine 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.**

Dabei sind der IRQ 10 sowie die Basisadressen wichtig.

0xb000: PLX Basis Adresse (PCI Interface, hier ohne Bedeutung)

0xa800: Uart Block Adresse (ab hier liegen die Schnittstellen)

0xa400: Interrupt Vektor Adresse, dazu später.

Der IRQ 10 wird bei **Setserial** eingetragen, die Schnittstellen liegen dann in Schritten von **8** fortlaufend ab der Basis Adresse. Diese Ressourcen werden bei jedem Neustart des PC dynamisch zugeteilt; sie ändern sich aber erst dann, wenn die sonstige Konfiguration des Rechners anders ist. Man kann sich also auf diese Werte verlassen.

Außerdem ist eine VScom PCI Karte automatisch auf die hohe Übertragungsrate eingestellt, also achtfach. Es würde also immer eine höhere Baudrate erreicht, als vom Programm konfiguriert. Um dieses zu korrigieren, benötigt **Setserial** eine zusätzliche Angabe, nämlich **baud\_base 921600**. Der Treiber wird so konfiguriert, daß er alle Geschwindigkeiten basierend auf der maximalen Übertragungsrate von 921.6kbps errechnet.

IRQ-Vektor bzw. fourport Option:

Die Karten der Serie VScom 400 und VScom 800 bieten ein gesondertes IRQ-Vektor (oder IRQ-Status) Register. Mit diesem Register läßt sich die Option des IRQ-Sharing wesentlich effektiver nutzen.

Technisch: das Register wird gebildet durch einen LS240 Treiber, an dessen Eingängen die IRQ-Ausgänge der UARTs liegen. Bei Abfrage werden die Werte auf den Datenbus gelegt. Hierbei entspricht Bit 0 dem ersten UART (COMA, Port 1), Bit 1f dem zweiten bis zum Bit 3 (bzw. Bit 7 auf der VScom 800). Im Ruhezustand ist dort eine „1“ zu lesen, im aktiven Zustand entsprechend eine „0“.

Dieses Register läßt sich nun nutzen, um die Bearbeitungszeit eines Interrupts drastisch zu verkürzen. In dem obigen Beispiel ist diese Möglichkeit mittels der Angabe **^fourport** explizit



## VS Vision Systems GmbH

Industrielle Bildverarbeitung

Aspeloh 27A; 22848 Norderstedt

Tel : +49 (40) 528 401-0

Fax : +49 (40) 528 401-99

eMail: Multi@visionsystems.hh.uunet.de

deaktiviert worden. Der Begriff **fourport** stammt von einer Schnittstellenkarte mit vier seriellen Anschlüssen (AST). Der Linux-Treiber versteht darunter aber alle Kombinationen von seriellen Schnittstellen, die ein gemeinsames IRQ-Vektor Register nutzen. Dieses wird mittels der Option **set\_multiport** aktiviert.

```
#{SETSERIAL} cua2 set_multiport port1 0xa400 mask1 0xf match1 0xf
```

Bei einer VScom 800 wäre mask1 dann 0xff, ebenso match1, um alle 8 Schnittstellen zu steuern.

### Kompatibilität und höhere Geschwindigkeiten:

Besonders ältere Programme erlauben nicht, alle Geschwindigkeiten (insbesondere hohe) anzugeben. Für diese Zwecke gibt es die Optionen `spd_hi` und `spd_vhi`. Wenn eine Schnittstelle mit dieser Angabe konfiguriert ist, dann verwendet der Treiber statt der Baudrate 38400 die Geschwindigkeit 57.6kbps (bzw. 115.2kbps). Diese Angabe folgt am Ende der jeweiligen Konfigurationszeile.

### Einstellung des Handshake:

Stty – setze terminal charakteristika

Wenn Ihr Modem den Hardware-Handshake unterstützt, sollten Sie sicherstellen, daß dieser auch aktiviert ist. Überraschenderweise versuchen die wenigsten Kommunikations-Programme dies einzustellen, weshalb Sie es unter Umständen per Hand tun müssen. Das geschieht am besten auch im serial-Script, indem Sie stty so aufrufen:

```
$ stty crtscts < /dev/cua1
```

Hier ein vollständiges Script für die VScom 400 PCI aus obigem Beispiel:

```
#{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
```

Die Karten der VScom Turbo – Serie erlauben verschiedene Maximalgeschwindigkeiten. Bei den Serien PRO und PCI sind die Faktoren 1 und 8 möglich. Je nach Konfiguration sind also verschiedene Werte für **Baud\_base** möglich:

	Faktor	Baud Base
<b>Alle Karten</b>	1	115,200 kbps
<b>VScom Turbo, VScom 200 PRO</b>	2	230,400 kbps
<b>VScom Turbo, VScom 200 PRO, VScom 100 PRO</b>	4	460,800 kbps
<b>VScom PRO, außer VScom 200 PRO, VScom 100 PRO</b>	8	921,600 kbps
<b>VScom 100 PRO</b>	12	1,382400 Mbps

Das Register IRQ-Vektor ist verfügbar auf den Karten VScom 400 und VScom 800 der PRO- und PCI-Serie; auf den Karten VScom 400 Turbo und VScom 800 Turbo nur im Unix-Modus.

Die obigen Einstellungen – insbesondere für die VScom 400 PCI – wurden getestet und übermittelt durch Herrn Stephan Insringhausen-Bley von der Ruhr Universität Bochum. An dieser Stelle noch einmal vielen Dank dafür.