



P4dragon
DR-7X00

Version 1.0
08 / 2012

Linux & P4dragon DR-7X00

Betrieb des **SCS** P4dragon DR-7X00 unter Linux

SCS Spezielle Communications Systeme GmbH & Co. KG



Zum Betrieb des **SCS** P4dragon DR-7X00 mit Linux gibt es zwei Möglichkeiten:

1.) Angepasstes Kernel-Modul (drivers/usb/serial/ftdi_sio.c):

Das angepasste Modul setzt beim Anschluss eines P4dragon DR-7X00 die spezielle Baudrate automatisch. In der Applikation wird dann einfach 38400 Baud eingestellt. 38400 Baud wird durch den Treiber auf die spezielle Baudrate des P4dragon DR-7X00 geändert.

Einen Patch für das Kernel-Modul finden Sie in der Datei SCS-ftdisio.patch. Getestet wurde der Patch mit Kernel 3.4.6 und Kernel 3.2.0 (Ubuntu 12.04).

2.) Möchte oder kann man das Kernel-Modul nicht selbst übersetzen so besteht auch die Möglichkeit, die Baudrate für den P4dragon direkt in der Anwendung zu setzen. Der folgende Code-Ausschnitt zeigt die Vorgehensweise. Eine vollständige Beispiel-Applikation finden Sie in term.zip.

```
/*
 * open the port with 38400 baud
 */

int port;
struct termios savedAttr, newAttr;
struct serial_struct sstruct;

port = open("/dev/ttyUSB0", O_RDWR | O_NOCTTY | O_NONBLOCK);

tcgetattr(port, &savedAttr); // save this to restore later
newAttr = savedAttr;
newAttr.c_cflag = B38400 | CS8 | CLOCAL | CREAD;
cfmakeraw (&newAttr);
tcsetattr (port, TCSANOW, &newAttr);

/*
 * set the special baudrate
 */

// get serial_struct
if (ioctl(port, TIOCGSERIAL, &sstruct) < 0)
{
    printf("Error: could not get comm ioctl\n");
    close (port);
    exit(0);
}

// set custom divisor to get 829440 baud
sstruct.custom_divisor = 29;
sstruct.flags |= ASYNC_SPD_CUST;

// set serial_struct
if (ioctl(port, TIOCSSERIAL, &sstruct) < 0)
{
    printf("Error: could not set custom comm baud divisor\n");
    close (port);
    exit(0);
}
```



Spezielle Communications Systeme GmbH & Co. KG
Röntgenstraße 36
63454 Hanau
GERMANY

Internet: www.p4dragon.com
E-Mail: info@p4dragon.com

Tel.: +49 (0)6181 85 00 00
Fax.: +49 (0)6181 99 02 38