P4dragon DR-7X00

Version 1.0
08 / 2012

Linux & P4dragon DR-7X00

Using the SCS P4dragon DR-7X00 with Linux

SCS Spezielle Communications Systeme GmbH & Co. KG

There are two ways to use the **SCS** P4dragon DR-7X00 with Linux:

1) Patched kernel module (drivers/usb/serial/ftdi_sio.c):
   The patched kernel module sets automatically the special baudrate for the P4dragon DR-7X00. Within the application you set the baudrate to 38400 baud wich will be matched to the P4dragon DR-7X00 special baudrate by the kernel module.

   The neccessary kernel patch you will find in the file SCS-ftdisio.patch.
   The patch was tested with kernel 3.4.6 and kernel 3.2.0 (Ubuntu 12.04).

2) If you don't like to compile your own kernel there is a way to set the special baudrate directly in the application. The following code snippet shows the general procedure. A complete example application is available 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);
    }
```

SCS

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

Internet: www.p4dragon.com
Email: info@p4dragon.com

FON: +49 (0)6181 85 00 00
FAX: +49 (0)6181 99 02 38