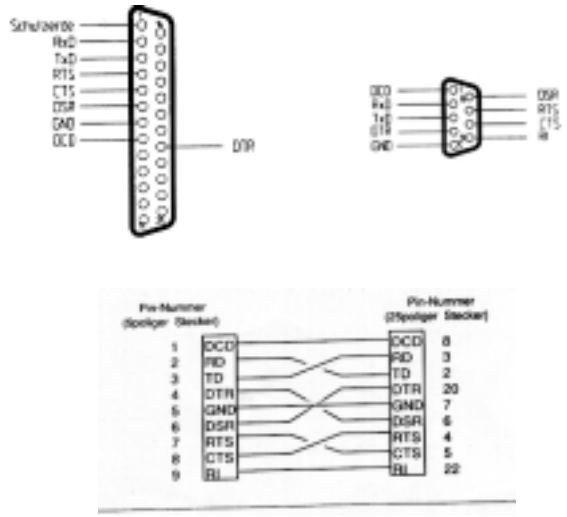
Hardwarenahe Programmierung in C/C++

Serielle Schnittstelle

Christian Zahler

Die Programmierung der seriellen Schnittstelle

V.24-Norm (europäisch) oder RS232c-Norm (USA – reference standard number 232 Version c) – praktisch idente Normen. Rrelativ niedrige Geschwindigkeiten, ungesicherte Übertragung; deshalb: max. 15 m lang



Port-Basisadresse 1. serielle Schnittstelle: 03f8/IRQ4
Port-Basisadresse 2. serielle Schnittstelle: 02f8/IRQ3
Port-Basisadresse 3. serielle Schnittstelle: 03e8/IRQ4
Port-Basisadresse 4. serielle Schnittstelle: 02e8/IRQ3

Die Informationslänge beträgt 8 Byte

03f8 = 03f8 + 00 ... Transmit and Receive buffer / Baud Rate
03f9 = 03f8 + 01 ... Interrupt enable register / Baud Rate
03fa = 03f8 + 02 ... Interrupt identification register
03fb = 03f8 + 03 ... LCR ... Line Control Register
03fc = 03f8 + 04 ... MCR ... Modem Control Register
03fd = 03f8 + 05 ... LSR ... Line Status Register
03fe = 03f8 + 06 ... MSR ... Modem Status Register

Baudratenbestimmung

Die Register 03f8 und 03f9 werden doppelt verwendet, einerseits als Sende-/Empfangspuffer bzw. als Interrupt-Enable-Register, andererseis zum Einstellen der Übertragungsgeschwindigkeit.

Wie die Register konkret verwendet werden, hängt vom Bit 7 des LCR ab!

Die Geschwindigkeit der Übertragung wird durch einen Quarzkristall bestimmt, der einen Takt von 119 kHz liefert. Damit ist eine maximale Übertragungsrate von 115200 Baud möglich. Die tatsächliche Baudrate wird durch den „Baudratenteiler" bestimmt. Dabei wird der Wert 115200 Baud (= Maximum) durch den Teiler dividiert.

Wenn das Bit 7 im LCR auf 1 gesetzt ist, gilt:

Der Baudratenteiler wird auf Adresse

03f8 + 00 ...unteres Byte des Teilers so- wie
03f8 + 01 .... oberes Byte des Teilers

gesetzt.

Wenn Bit 7 im LCR auf 0 gesetzt wird, haben die Register die „ursprüngliche Funktion":

Die Leitungen der V.24-Schnittstelle

Name der Leitung	Abkürzung	engl. Name	Position am 25poligen Stecker	Position am 9poligen Stecker
Schutzerde	GND	Ground	1	
Sendedaten	TD	Transmitted Data	2	3
Empfangsdaten	RD	Received Data	3	2
Sendeanforderung	RTS	Request to Send	4	7
Sendebereitschaft	CTS	Clear to Send	5	8
Betriebsbereitschaft	DSR	Data Set Ready	6	6
Signalerde	GND	Signal Ground	7	5
Empfangssignal	(D)CD	(Data) Carrier Detect	8	1
Testspannung +			9	
Testspannung –			10	
Hohe Sendefrequenzlage einschalten			11	
Empfangssignalpegel			12	
2. Sendebereitschaft			13	
2. Sendedaten			14	
Sendeschrittakt	TC		15	
2. Empfangsdaten			16	
Empfangsschrittakt	RC		17	
Nicht definiert			18	
2. Sendeanforderung			19	
Betriebsbereit	DTR	Data Terminal Ready	20	4
Empfangsqualität	SQ		21	
Rufsignal	RI	Ring Indicator	22	9
Hohe Übertragungsgeschwindigkeit einschalten			23	
2. Sendeschrittakt			24	
Nicht definiert			25	

03f8 + 00 Sende-/Empfangsdatenregister
03f8 + 01 Interrupt enable register

Beispiel: Eine Baudrate von 9600 Baud = 115200 / 12 soll eingestellt werden. Daher muss der binäre Wert 12 (Hex: 0C) auf die Adresse 03f8 geschrieben werden:

0	0	0	0	1	1	0	0

Ansteuerung der seriellen Schnittstelle mit inportb/outportb

```
// Programmname:
// Schnittstellenansteuerung
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <process.h>
#include <stdlib.h>
/*---------------- Konstante ----------------*/
#define COM1   0x3f8
#define COM2   0x2f8
#define COM3   0x3e8
#define COM4   0x2e8
#define LSR     0x05
#define LCR     0x03
/*----------Globale Variablen ----------*/
int     bit;
int     x;   // Koordinate für Sende-/Empfangszeichen
char   empfang;  // empfangenes Zeichen
char   send;     // gesendetes Zeichen

void SetBit (unsigned short byte,unsigned short bitnr)
{
 unsigned short Register;
 Register=inportb(byte);
 outport(byte,Register||bitnr);
}

void ClearBit (unsigned short com_port, unsigned short bitNr)
{
 outportb(com_port,inportb(com_port)&~bitNr);
}

void Init V24 ()
{
SetBit (COM1+LCR,1); SetBit (COM1+LCR,2);  // 3 Datenbits
ClearBit (COM1+LCR,3); // 0 Stopbits
ClearBit (COM1+LCR,4); // no parity enable
ClearBit (COM1+LCR,5); // parity odd

outportb (COM1+LCR,131);
// Divisor Latch
outportb (COM1,12);         //  115200 / 12 = 9600 Baud
Übertragungsrate
outportb (COM1+LCR,3);           // zurückstellen auf Daten
}

void Empfangen (void)
{
empfang = inportb(COM1);
gotoxy (x,10);
printf("%c",empfang);
}

void Senden (void)
{
send = getch();
gotoxy(x,4);
printf ("%c",send);
outportb(COM1,send);
}

void main (void)
{
Init V24 ();
ClearBit (COM1+LSR,1);
do
  {
  bit = inportb(COM1+LSR)&1;

  if (bit == 1)
    {
    x++; Empfangen ();
    }
  else
    {
    if (kbhit() != 0)
      {
      x++; Senden ();
      }
    else
      {
      ;   // wenn keine Taste gedrückt, nichts machen
      }
    }
  }while (send != 13);
textcolor (LIGHTGRAY);
clrscr ();
}
```

Ansteuerung der seriellen Schnittstelle mit inportb/outportb (Vollversion)

```
// Programmname:
//   Schnittstellenansteuerung
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <process.h>
#include <stdlib.h>
/*---------------- Konstante ----------------*/
#define COM1   0x3f8
#define COM2   0x2f8
#define COM3   0x3e8
#define COM4   0x2e8
#define DTR  0x01
#define BITNR  1
#define OFF  0x00
#define ON   0x01
#define MCR  0x04
#define LSR     0x05
#define LCR     0x03

/*----------Globale Variablen ----------*/
int  taste;
int  i;
int bit;
char   empfang;
char   send;
int  ex,ey;
int  sx,sy;
int db,sb,pe,ps,br,pt;
int  com port;
int lcrbyte;

/*----------------------------------------*/
void SetBit (unsigned short byte,unsigned short bitnr)
{
 unsigned short Register;
 Register=inportb(byte);
 outport(byte,Register||bitnr);
}
/*----------------------------------------*/
void ClearBit (unsigned short com_port, unsigned short bitNr)
{
 outportb(com_port,inportb(com_port)&~bitNr);
}
/*----------------------------------------*/
void Rahmen (int links,int oben,int breite,int hoehe,int
strichfarbe,int hintergrund,char *p header)
{
textcolor (strichfarbe);
textbackground (hintergrund);

gotoxy (links,oben);
for (i=1; i<=breite; i=i+1)
  {
  cprintf ("Í");
  gotoxy (links+i,oben);
  }
gotoxy (links,oben+hoehe);
for (i=1; i<=breite; i=i+1)
  {
  cprintf ("Í");
  gotoxy (links+i,oben+hoehe);
  }
gotoxy (links,oben);
for (i=1; i<=hoehe; i=i+1)
  {
  cprintf ("º");
  gotoxy (links,oben+i);
  }
for (i=1; i<=hoehe; i=i+1)
  {
  cprintf ("º");
  gotoxy (links+breite,oben+i);
  }
gotoxy (links,oben); cprintf ("É");
gotoxy (links,oben+hoehe); cprintf ("È");
gotoxy (links+breite,oben); cprintf ("»");
gotoxy (links+breite,oben+hoehe); cprintf ("¼");

gotoxy (links+(breite/2-strlen(p_header)/2),oben);
cprintf (p header);

textcolor (LIGHTGRAY);
textbackground (BLACK);
}       // Rahmen
```

```
/*-------------------------------------*/
void Maske (void)
{
clrscr();
Rahmen(1,1,79,11,YELLOW,BLACK," Senden ");
Rahmen(1,13,79,11,YELLOW,BLACK," Empfangen ");
textcolor (LIGHTGRAY);
gotoxy (3,2);
}
/*-------------------------------------*/
void Menu ()
{
do
  {
  clrscr();
  Rahmen(1,1,79,23,GREEN,BLACK," Men  ");
  textcolor (LIGHTGRAY);
  gotoxy (4,4); cprintf ("Port:  ");
  gotoxy (4,5); cprintf ("1........COM1");
  gotoxy (4,6); cprintf ("2........COM2");
  gotoxy (4,7); cprintf ("3........COM3");
  gotoxy (4,8); cprintf ("4........COM4");
  gotoxy (30,4);cprintf ("Baudrate:    ");
  gotoxy (30,5);  cprintf ("192..... 600 Baud");
  gotoxy (30,6);  cprintf ("96.......1200 Baud");
  gotoxy (30,7);  cprintf ("48.......2400 Baud");
  gotoxy (30,8);  cprintf ("24.......4800 Baud");
  gotoxy (30,9);  cprintf ("12.......9600 Baud");
  gotoxy (30,10);   cprintf ("6.......19200 Baud");
  gotoxy (56,4);  cprintf ("Anzahl der Datenbit:  ");
  gotoxy (56,5);  cprintf ("0........5 Datenbit");
  gotoxy (56,6);  cprintf ("1........6 Datenbit");
  gotoxy (56,7);  cprintf ("2........7 Datenbit");
  gotoxy (56,8);  cprintf ("3........8 Datenbit");
  gotoxy (4,15);  cprintf ("Anzahl der Stopbit:  ");
  gotoxy (4,16);  cprintf ("0........1 Stopbit");
  gotoxy (4,17);  cprintf ("1........2 Stopbit");
  gotoxy (30,15);   cprintf ("Parity Enable:  ");
  gotoxy (30,16);   cprintf ("0........No Parity");
  gotoxy (30,17);   cprintf ("1........Enable Parity");
  gotoxy (56,15);   cprintf ("Parity Select:  ");
  gotoxy (56,16);   cprintf ("0........Odd Parity");
  gotoxy (56,17);   cprintf ("1........Even Parity");
  do
    {
    gotoxy (10,4); printf ("    ");
    gotoxy (10,4); scanf ("%d",&pt);
    }while (pt != 1 && pt != 2 && pt !=3 && pt !=4);
  if (pt == 1)
    com port = COM1;
  if (pt == 2)
    com port = COM2;
  if (pt == 3)
    com port = COM3;
  if (pt == 4)
    com port = COM4;
  do
    {
    gotoxy (40,4); printf ("___ ");
    gotoxy (40,4); scanf ("%d",&br);
    }while (br != 6 && br != 12 && br != 24 && br != 48 && br != 96 &&
br != 192);
  outportb (com_port+LCR,131);
// Divisor Latch
  outportb (com_port,br);           // Teiler (96=1200Baud) (115200 /
Baudrate)
  outportb (com_port+LCR,3);        // zurückstellen auf Daten
  do
    {
    gotoxy (77,4); printf ("  ");
    gotoxy (77,4); scanf ("%d",&db);
    }while (db != 3 && db != 2 && db != 1 && db !=0);
  if (db == 3)
    SetBit (com port+LCR,1); SetBit (com port+LCR,2);
  if (db == 2)
    SetBit (com port+LCR,2);
    ClearBit (com_port+LCR,1);
  if (db == 1)
    SetBit (com_port+LCR,1);
    ClearBit (com port+LCR,2);
  do
    {
    gotoxy (24,15); printf ("_   ");
    gotoxy (24,15); scanf ("%d",&sb);
    }while (sb != 1 && sb != 0);
  if (sb == 0)
    ClearBit (com port+LCR,3);
  if (sb == 1)
    SetBit (com_port+LCR,3);
  do
    {
    gotoxy (45,15); printf ("    ");
    gotoxy (45,15); scanf ("%d",&pe);
    }while (pe != 1 && pe != 0);
  if (pe == 0)
    ClearBit (com port+LCR,4);
  if (pe == 1)
    SetBit (com port+LCR,4);
  do
    {
    gotoxy (71,15); printf ("_   ");
    gotoxy (71,15); scanf ("%d",&ps);
    }while (ps != 1 && ps != 0);
  if (ps == 0)
    ClearBit (com_port+LCR,5);
  if (ps == 1)
    SetBit (com_port+LCR,5);
  gotoxy (25,21); printf ("Einstellungen korrekt N ?");
  taste = getch ();
  }while (taste != 'j');
}

/*-------------------------------------*/
void Init_V24 ()
{
if (pt == 1)
  com_port = COM1;
if (pt == 2)
  com port = COM2;
outportb (com_port+LCR,131);
// Divisor Latch
outportb (com_port,br);             // Teiler (96=1200Baud) (115200 /
Baudrate)
outportb (com_port+LCR,3);          // zur ckstellen auf Daten
}
/*-------------------------------------*/
void Empfangen (void)
{
empfang = inportb(com_port);
gotoxy (ex,ey);
printf("%c",empfang);
if (ey >= 24)
  {
  ex=3;ey=14;
  for (i=14;i<=4;i++)
    {
    gotoxy (1,i); clreol ();
    }
  Rahmen(1,13,79,11,YELLOW,BLACK," Empfangen ");
  gotoxy (ex,ey);
  }
else
  {
  if (ex <=77)
    {
    ex++;
    }
  else
    {
    ex = 3;
    ey++;
    }
  }
}
/*-------------------------------------*/
void Senden (void)
{
send = getch();
gotoxy (sx,sy);
printf ("%c",send);
outportb(com port,send);
if (sy >= 12)
  {
  sx=3;sy=2;
  for (i=2;i<12; i++)
    {
    gotoxy (1,i); clreol ();
    }
  Rahmen(1,1,79,11,YELLOW,BLACK," Senden ");
  gotoxy (sx,sy);
  }
else
  {
  if (sx <= 77 )
    {
    sx++;
    }
  else
```

```
      {
      sx = 3;
      sy++;
      }
    }
  }

  /*-------------------------------------------*/
  int Isbit (unsigned short com_port, unsigned short bitNr)
  {
   return(inportb(com_port)&bitNr);
  }
  /*-------------------------------------------*/

  void setRTS (int port,char ea)
  {
   outportb(port,0 );
   if(ea == ON)
    outportb(MCR,3);
   if(ea == OFF)
   outportb(MCR,0);
  }
  /*-------------------------------------------*/

  void getCTS (int port)
  {
  outportb(port,0);
  inportb(MCR);
  }
  /*-------------------------------------------*/

  void main (void)
  {
  Menu ();
  Init_V24 ();
  ClearBit (com port+LSR,1);
  Maske ();
  ex=3;sx=3;ey=14;sy=2;
  do
    {
    bit = Isbit (com_port+LSR,1);
    if (bit == 1)
      {
      Empfangen ();
      }
    else
      {
      if (kbhit() != 0)
        {
        Senden ();
        }
      else
        {
        ;
        }
```

```
      }
    }while (send != 13);
  textcolor (LIGHTGRAY);
  clrscr ();
  }
```

Die Artikelserie “Hardwarenahes Programmieren in C und C++” wurde mit dieser Folge abgeschlossen. Die restlichen Beiträge dieser Beitragserie finden Sie in den Ausgaben **PCNEWS**-63, 67, 68, 69 und 70.