

RS232  
Protokol z předmětu 36PZ

Miloš Hrdý

Martin Bruchanov

Tomáš Uko

Jiří Trávníček

23. dubna 2006

# 1 Zadání

Sestavte program pro komunikaci po sériové lince a to přímým programováním registrů asynchronního obvodu UART 8250 dle následujících pokynů:

- napište program pro vysílač znaků po sériové lince,
- napište program pro přijímač znaků přicházejících po sériové lince,
- programy odladte a předvedte funkčnost na propojené dvojici počítačů PC; přijímané a vysílané znaky zobrazujte v oddělených částech obrazovky monitoru, prozkoumejte chování obvodu 8250 při různých rychlostech.

*Poznámka:* Programy nejprve napište bez použití přerušení, až poté s využitím služeb přerušovacího systému.

# 2 Vypracování

**Pomůcky:**

1. 2 ks PC x86 kompatibilní
2. Kabel pro propojení přes sériový port
3. Operační systém MS-DOS
4. Vývojové prostředí Turbo Pascal

Použili jsme předpřipravené programy `intersta.pas`, `seriasta.pas` pro obsluhu sériového portu, ve kterých jsme provedli několik úprav:

**seriasta.pas**

```
program serial;
uses dos, crt;
const
{----- COM ports in PC -----}
COM1= $3f8;
COM2= $2f8;
Intr: Integer = $0b;

:

procedure init_com (portaddr: word);
begin
port[portaddr+LCR ]:=$80;      { <- $3FB přepnout na nastavení rychlosti }
port[portaddr + DLM]:=DLM_HIGH;  { > 300 Baud }
port[portaddr + DLL]:=DLL_1200;  { 1200 Baud }
port[portaddr+LCR]:=0;        { $3FB přepnout zpět }
port[portaddr + LCR]:=LCR_8BIT;  { Line Control Register, 8 bit, bez parity }
port[portaddr + IER]:=0;      { bez přerušení }
port[portaddr + MCR]:=MCR_DTR;  { Modem Control Register: DTR}
end;

function rx_rdy (portaddr: word): boolean;
begin
rx_rdy:=((port[portaddr + LSR] and 1) <> 0); { <- maska }
end;

function tx_rdy (portaddr: word): boolean;
begin
tx_rdy:=((port[portaddr + LSR] and LSR_BF) <> 0); { <- registr }
end;

:
```

## intersta.pas

```
program interp;
uses dos, crt;
const
  {----- COM ports in PC -----}
  COM1= $3f8;
  COM2= $2f8;
  Intr: Integer = $0c; {x0c - COM1, x0b - COM2}

:
:

var
  com: word;
  intno: byte;      { číslo přerušeni }
  oldvect: pointer; { pro uloženi původního vektoru přerušeni }
  c: char;
  xp, yp ,xv ,yv: byte;

:
:

procedure rx_int; interrupt; { obsluha přerušeni }
begin
  window(1, 13, 80, 25);
  gotoxy(xp, yp);
  c:= rx_get(com);      { načti znak }
  write(c);            { vypiš }
  if c = chr(13) then
    writeln;
  xp:= wherex;
  yp:= wherey;
  port[$20] := $20;    { odhlášení řadiči přerušeni }
end;

begin                { hlavní program }

:
:

  init_com(com);      { inicializace }
  GetIntVec(intno,oldvect); { uloženi vektoru přerušeni }
  SetIntVec(intno,@rx_int); { nastavení vlastní obsluhy }
  while (ord(c) <> 27) do { dokud nebyl stisknut Esc }
  begin
    if keypressed then
      begin
        if tx_rdy(com) then { je připraven pro vysílání }
          begin
            window(1, 3, 80, 10);
            gotoxy(xv, yv);
            c:=readkey;      { přečti znak z klávesnice }
            tx_put(com, c);  { odešli }
            write(c);        { vypiš na obrazovku }
            if c = chr(13) then { Enter }
              writeln;
            xv:= wherex;
            yv:= wherey;
          end;
        end;
      end;
    end;
  end;
  c:=readkey;           { ještě počká na 1 stisk klávesy }
  if c = #0 then        { rozšířený kód }
    c:=readkey;         { přečti, aby se po ukončení nepoužila tato klávesa }
  SetIntVec(intno,oldvect); { vrácení původního vektoru }
end.
```

### 3 Závěr

Během testování programů jsme se setkali jen s menšími problémy, když nám náhle přestal fungovat již odladěný program–pomohl restart. Při bližším pohledu do zdrojových souborů programů je vidět, že je potřeba nastavit mnoho registrů (`procedure init_com`), aby se podařilo zprovoznit sériový port.