PERCOBAAN 10 PORT SERIAL

TUJUAN

- 1. Memahami rangkaian interface port serial
- 2. Memahami mengkonfigurasikan komunikasi serial pada port serial PC
- 3. Memahami penulisan bahasa assembly untuk konfigurasi port serial
- 4. Memahami penulisan pemrograman delphi untuk komunikasi dengan mikrokontroller.

Salah satu komunikasi yang paling ampuh untuk diimplementasikan dalam sistem komunikasi digital adalah komunikasi dengan memanfaatkan jalur serial RS232. Mikrokontroller 89S52 telah memiliki fasilitas UART, sehingga dapat melakukan komunikasi secara serial dengan level RS2322 antar peralatan atau dengan komputer. MAX232 merupakan IC yang difungsikan untuk merubah format TTL ke RS232 atau sebaliknya.



Gambar 10.1 Sistem komunikasi serial mikrokontroller dengan PC



Gambar 10.2. Rangkaian skematik MAX232

Konfigurasi komunikasi serial dengan Hyperterminal

Pengujian sistem komunikasi antara mikrokontroller dengan port serial computer dapat dilakukan dengan memanfaatkan program hyperterminal pada sistem yang berbasis under windows (98, ME dan XP). Kelebihan penggunakan program ini, tidak diperlukan suatu bahasa pemrograman dan kelemahannya, terbatas hanya pada proses pengujian.. Adapun cara mengkonfigurasikannya adalah sebagai berikut:

Gambar 10.2 Pemilihan menu HyperTerminal pada communication

Gambar 10.3. (a) Pemberian nama koneksi (b) Pemilihan koneksi pada COM1

Gambar 10.4. (a) Properties serial komunikasi (b) windows HyperTerminal

Percobaan 10.1.

Mengambil Data Karakter dari ROM dan Display ke Monitor dengan Hyperterminal.

- 1. Hubungkan kabel serial dari board Trainer ke Port Comm PC
- 2. Buat komunikasi dengan Hyperterminal
- 3. Ketik program asembly berikut ini:

```
org Oh
      nop
             initserial
      call
      mov dptr, #MyCharacter
gets: clr
             Α
      Movc A,@A+dptr
      cjne
            A,#0,Next
      sjmp getout
Next: call sendout
      inc
             dptr
      sjmp
             gets
getout:sjmp
            getout
      ;
initserial:
      mov scon,#52h ; Konfigurasi komunikasi serial mode 1
      mov tmod, #20h ; Baud rate 2400 BPS
      mov th1,#0F3H
      setb tr1
      ret
inchar:
detecti: jnb ri,detecti ;DeteksibitRIapadatasudah diterima belum
      clr
             ri
            a,sbuf
      mov
      Subb A,#30h
      Mov
             R0,A
      ret
Sendout:
detecto: jnb ti,detecto;
             ti ;
      clr
             sbuf,a ;
      mov
      ret
MyCharacter:
      DB ' Selamat Datang Di Pelatihan Mikrokontroller',13,10,0
      End
```

- 4. Simpanlah program yang anda ketik dan beri nama : hiper1a.asm
- 5. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari *.asm ke *.hex.
- 6. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software (Lihat Petunjuk Penggunaan)
- 7. Lakukan pengamatan pada Layar Hyper Terminal

Percobaan 10.2. Pengendalian LED dengan menggunakan Hyperterminal

Pada percobaan ini LED dihidupkan sesuai dengan karakter (1 –2 - 4) yang diterima dari keyboard PC.

- 1. Hubungkan jumper konektor ke LED_EN
- 2. Hubungkan modul Microcontroller Trainer dengan power supply +5V
- 3. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
- 4. Buka Program Hyperterminal dan inputkan angka 0 s/d 9
- 5. Ketik program asembly berikut ini:

	org	Oh
	nop	
	call	initserial
	mov	dptr,#MyCharacter
gets:	clr	A
	Movc	A,@A+dptr
	cjne	A,#0,Next
	sjmp	getin
Next:	call	sendout
	inc	dptr
	sjmp	gets
getin:	call	inchar
	cjne	A,#'1',next1
	mov	P0,#1111110b
next1:	cjne	A,#'2',next2
	mov	P0,#1111101b
next2:	cjne	A,#'4',next3
	mov	P0,#11111011b
next3:	sjmp	getin
	;	
initse	rial:	
	mov	scon,#52h ; Konfigurasi komunikasi serial mode 1
	mov	tmod,#20h ; Baud rate 2400 BPS
	mov	th1,#0F3H
	setb	trl
	ret	
inchar	:	
detect	i:jnb	ri,detecti;DeteksbitRIapadata sudah diterima belum
	clr	ri
	mov	a,sbuf
	ret	
Sendou	t:	
detect	o:jnb	ti,detecto;
	clr ti	;
	mov sb	uf,a ;
	ret	
MyChara	acter:	
	DB ' T	ekan Tombol 1, 2 atau 4',13,10,0
	End	
5. Laku	kan penel	kanan tombol 1, 2 atau 4

Percobaan 10.3 Pengiriman data secara serial dari PC ke mikrokontroller dengan output LED (pemrograman delphi)

Pada percobaan ini, data dikirimkan melalui port communication serial PC secara serial dan ditangkap oleh mikrokontroller untuk dioutputkan ke LED



- 1. Hubungkan jumper konektor ke LED_EN
- 2. Hubungkan modul Microcontroller Trainer dengan power supply +5V
- 3. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
- 4. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
- 5. Ketik program asembly berikut ini:

```
org
              0h
       nop
              initserial
       call
       call
              inchar
gets:
              P0,a
       mov
       sjmp
              gets
       ;
initserial:
              scon,#52h ; Konfigurasi komunikasi serial mode 1
       mov
              tmod,#20h ; Baud rate 2400 BPS
       mov
              th1,#-13
       mov
       setb
              tr1
       ret
       ;
inchar:
detect:jnb ri,detect;Deteks bit RI apadatasudah diterima belum
       clr ri
       mov a, sbuf
       ret
       ;
       End
```

- 6. Buka Program Delphi
- 7. Ketik Program Delphi berikut ini, untuk menjalankan komunikasi melalui port serila dengan base address 3F8h.

烤 Tes	ting		-		
0	1	2	4	8	
16	32	64	128	255	
File: testingLED.asm					

```
var
Form1: TForm1;
data,status : byte;
const
       base = $3f8;{base address port serial}
       lcr = 3; {line control register}
       dll = 0; {divisor lacht low byte}
       dlh = 1; {divisor lacht high byte}
       lsr = 5; {line status register}
implementation
  {$R *.DFM}
Procedure Initserial;
begin
       asm
       mov dx,base+lcr; {address line control register}
       mov al,$80 ; {1000000b = access bit divisor lacht}
       out dx,al
       ;
       mov dx,base+dll; {address divisor lacht low byte}
       mov al,$30 ; {DLLB = 30h}
       out dx,al
       ;
       mov dx,base+dlh; {address divisor lacht high byte}
       mov al, $00 ; {DLLH = 00h}
       out dx,al
       ; {Pada saat ini Port serial}
       ; {memp.baud rate = 2400 bps}
       mov dx,base+lcr; {address line control register}
       mov al,$03 ; {00000011b =}
       out dx,al ; {bit 7=0, access to Rx buffer & Tx
       ; {bit 6=0, set break disable
       ; {bit 5-4-3=000, no parity
       ; {bit 2=0, one stop bit
       ; {bit 1-0=11,data lenght 8 bit}
       end;
       end;
Procedure Send_Data_Serial;
begin
       asm
       mov dx, base
       mov al, data
```

```
out dx,al
       end
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
       Initserial;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
   repeat
       asm
       mov dx,base+lsr ; {address line status register }
       in al.dx
       and al,$20 ; {00100000b =not masking bit 5}
       mov status, al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
       data:=1;
       Send_Data_Serial;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
   repeat
       asm
       mov dx,base+lsr ; {address line status register }
       in al,dx
       and al,$20 ; {00100000b =not masking bit 5}
       mov status,al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
       data:=2;
       Send_Data_Serial;
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
   repeat
       asm
       mov dx,base+lsr ; {address line status register }
       in al,dx
       and al,$20 ; {00100000b =not masking bit 5}
       mov status, al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
      data:=4;
       Send_Data_Serial;
end;
procedure TForm1.Button4Click(Sender: TObject);
begin
   repeat
       asm
       mov dx,base+lsr ; {address line status register }
       in al,dx
       and al, $20 ; {00100000b =not masking bit 5}
       mov status,al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
```

```
data:=8;
       Send_Data_Serial;
end;
procedure TForm1.Button5Click(Sender: TObject);
begin
   repeat
       asm
       mov dx,base+lsr ; {address line status register }
       in al,dx
       and al, $20 ; {00100000b =not masking bit 5}
       mov status,al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
       data:=16;
       Send Data Serial;
end;
procedure TForm1.Button6Click(Sender: TObject);
begin
   repeat
       asm
       mov dx,base+lsr ; {address line status register }
       in al,dx
       and al,$20 ; {00100000b =not masking bit 5}
       mov status, al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
       data:=32;
       Send_Data_Serial;
end;
procedure TForm1.Button7Click(Sender: TObject);
begin
   repeat
       asm
      mov dx,base+lsr ; {address line status register }
       in al,dx
       and al, $20 ; {00100000b =not masking bit 5}
       mov status,al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
       data:=64;
       Send_Data_Serial;
 end;
procedure TForm1.Button8Click(Sender: TObject);
begin
   repeat
       asm
       mov dx,base+lsr ; {address line status register }
       in al,dx
       and al,$20 ; {00100000b =not masking bit 5}
       mov status,al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
       data:=128;
       Send_Data_Serial;
end;
```

```
procedure TForm1.Button9Click(Sender: TObject);
begin
   repeat
       asm
       mov dx,base+lsr ; {address line status register }
       in al,dx
       and al,$20 ; {00100000b =not masking bit 5}
       mov status,al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
       data:=0;
       Send_Data_Serial;
end;
procedure TForm1.Button10Click(Sender: TObject);
begin
   repeat
       asm
       mov dx,base+lsr ; {address line status register }
       in al,dx
       and al,$20 ; {00100000b =not masking bit 5}
       mov status,al ; {bit5=Empty Transmitting holding reg}
       end;
   until status = $20; { If ETHR = 1 then data ready tobe send }
       data:=255;
       Send_Data_Serial;
end;
end.
```

- 8. Simpanlah program yang anda ketik dan beri nama : prog101a.asm
- 9. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari *.asm ke *.hex.
- 10. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software (Lihat Petunjuk Penggunaan)
- 11. Jalankan program delphi dengan menekan tombol F9 atau RUN
- 12. Lakukan penekanan pada tiap-tiap button dan lakukan pengamatan.

Button	Keadaan LED (Nyala/ Padam)
1	
2	
4	
8	
16	
32	
64	
128	
255	

Percobaan 10.4

Pengiriman data secara serial dari ADC ke Mikrokontroller ke PC. (dengan pemrograman delphi)

Pada percobaan ini, data ADC0804 diambil menggunakan port 1 mikrokontroller dan dikirimkan melalui port serial ke PC



- 1. Hubungkan modul Microcontroller Trainer dengan power supply +5V
- 2. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
- 3. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
- 4. Ketik program asembly berikut ini:

```
org
               0h
       call
               initserial
                       ; ambil data dari adc
start: call
               ADC
       call
               Sendout
       sjmp
               start
       ;
               P3.3
ADC:
       clr
       nop
       setb
               P3.3
       jb
               P3.2,eoc
eoc:
               P3.4
       clr
       mov
               A,P1
```

```
setb P3.4
         ret
         ;
Sendout:
detect:jnb
                 ti,detect;
                  ti ;
         clr
                  sbuf,a ;
         mov
         ret
         ;
initserial:
                  scon,#52h ;initialize serial mode 1
tmod,#20h ;timer1 mode 2
th1,#0F3h ;Reload value for baud rate 2400
         mov
         mov
         mov
         setb
                  tr1
         ret
         end
```

- 5. Buka Program Delphi
- 6. Ketik Program Delphi berikut ini, untuk menjalankan komunikasi melalui port serial dengan base address 3F8h.

📌 Form1 📃 🗖 🗙
Receive Data Serially From ADC Data 100 Decimal
Var Forml: TForml:
data, status: byte; const
<pre>base = \$3f8;{base address port serial} lcr = 3; {line control register} dll = 0; {divisor lacht low byte} dlh = 1; {divisor lacht high byte} lsr = 5; {line status register}</pre>
implementation
{\$R *.DFM}
Procedure Initserial;
asm
<pre>mov dx,base+lcr; {address line control register} mov al,\$80 ; {10000000b = access bit divisor lacht} out dx,al ;</pre>
mov dx,base+dll; {address divisor lacht low byte} mov al,\$30 ; {DLLB = 30h} out dx,al ;
<pre>mov dx,base+dlh; {address divisor lacht high byte} mov al,\$00 ; {DLLH = 00h}</pre>

```
out dx,al
       ; {Pada saat ini Port serial}
       ; {memp.baud rate = 2400 bps}
       mov dx,base+lcr;{address line control register}
       mov al,$03 ; {00000011b =}
       out dx,al ; {bit 7=0, access to Rx buffer & Tx
       ; {bit 6=0, set break disable
       ; {bit 5-4-3=000, no parity
       ; {bit 2=0, one stop bit
       ; {bit 1-0=11,data lenght 8 bit}
       end;
end;
Procedure Receive_Data_Serial;
begin
       asm
      mov dx, base
       in al,dx
      mov data,al
       end
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
    Repeat
       asm
      mov dx,base+lsr ; { address line status register }
      in al,dx
       and al, $01 ; {LSR = 00000001b, detects bit 0}
       mov status,al ; {bit 0 = data ready}
       end;
    until status = $01;{ jika bit 0 = 1 then data ready}
      Receive Data Serial;
       edit1.text:=inttostr(data);
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
       initserial;
       timer1.enabled:=true;
end;
end.
```

- 7. Simpanlah program yang anda ketik dan beri nama : prog102a.asm
- 8. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari *.asm ke *.hex.
- Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software (Lihat Petunjuk Penggunaan)
- 10. Jalankan program delphi dengan menekan tombol F9 atau RUN
- 11. Lakukan pengamatan untuk data ADC yang ditampilkan.