

# PC-Link

---

## SERIAL PPI

### Trademarks & Copyright

AT, IBM, PC, and PC-DOS are trademarks of International Business Machines Corporation.

MS-DOS and Windows are registered trademarks of Microsoft Corporation.

Pentium is a registered trademark of Intel Corporation.

Borland Delphi is a copyright of Inprise Corporation.

Turbo Pascal is a copyright of Borland International Incorporated.

<i><b>PC-LINK</b></i>					
<i>Serial (COM) Port</i>	<i>Parallel (LPT) Port</i>	<i>USB</i>	<i>Firewire</i>	<i>ISA slot</i>	<i>PCI slot</i>
✓					

# Daftar Isi

---

<b>1.</b>	<b>Pendahuluan.....</b>	<b>1</b>
1.1.	Spesifikasi Eksternal SERIAL PPI.....	1
1.2.	Spesifikasi Internal SERIAL PPI.....	1
1.3.	Sistem yang Dianjurkan.....	1
<b>2.</b>	<b>Perangkat Keras SERIAL PPI.....</b>	<b>2</b>
2.1.	Tata Letak Komponen SERIAL PPI.....	2
2.2.	Alokasi dan Spesifikasi Port.....	2
2.3.	Hubungan SERIAL PPI dengan Komputer.....	3
2.4.	Mencoba SERIAL PPI dengan SERPPI.EXE.....	3
2.5.	Mencoba SERIAL PPI dengan PASPPI.EXE.....	3
<b>3.</b>	<b>Perangkat Lunak SERIAL PPI.....</b>	<b>3</b>
3.1.	Spesifikasi UART RS-232.....	3
3.2.	Command.....	4
3.2.1.	Byte Transfer.....	4
3.2.2.	Bit Set/Reset.....	5
3.2.3.	Counter.....	7
3.2.4.	Baud Rate.....	8
3.3.	Rutin DLL dan TPU.....	8
3.4.	Contoh Aplikasi dan Program.....	13
3.5.	Kerangka Program.....	15
	<b>Lampiran</b>	
A.	Skema SERIAL PPI.....	17
B.	Protokol SERIAL PPI.....	18

## 1. PENDAHULUAN

PC-Link SERIAL PPI merupakan pengendali 40 bit jalur input/output melalui antarmuka UART RS-232 yang dapat dihubungkan ke komputer secara langsung. Contoh aplikasi dari SERIAL PPI adalah sebagai pengendali tampilan LED, sebagai pembaca kondisi saklar, penghitung pulsa counter, dan lain-lain.

### 1.1. SPESIFIKASI EKSTERNAL SERIAL PPI

Spesifikasi Eksternal SERIAL PPI adalah sebagai berikut:

- Menggunakan antarmuka UART RS-232.
- 4 pilihan Baud Rate.
- 16 bit jalur Input/Output (Port 1 dan Port 2) dengan level CMOS.
- 24 bit jalur Programmable Peripheral Interface 82C55 (Port A, Port B, dan Port C) dengan level CMOS.
- 2 Counter 16 bit (Counter 0 dan Counter 1) dengan level CMOS.
- Sumber tegangan input 12 VDC.
- Tersedia Voltage Regulator dengan tegangan output 5 VDC.

### 1.2. SPESIFIKASI INTERNAL SERIAL PPI

Dalam penggunaan dari UART SERIAL PPI dikenal adanya Protocol Layer.

**UART Protocol Layer** adalah lapisan yang dipergunakan untuk mengatur semua lalu lintas data dan sudah tersusun sesuai kegunaan menjadi paket Sub-rutin.

Adapun daftar API Command terdapat pada **bagian 3.2**.

### 1.3. SISTEM YANG DIANJURKAN

Perangkat Keras:

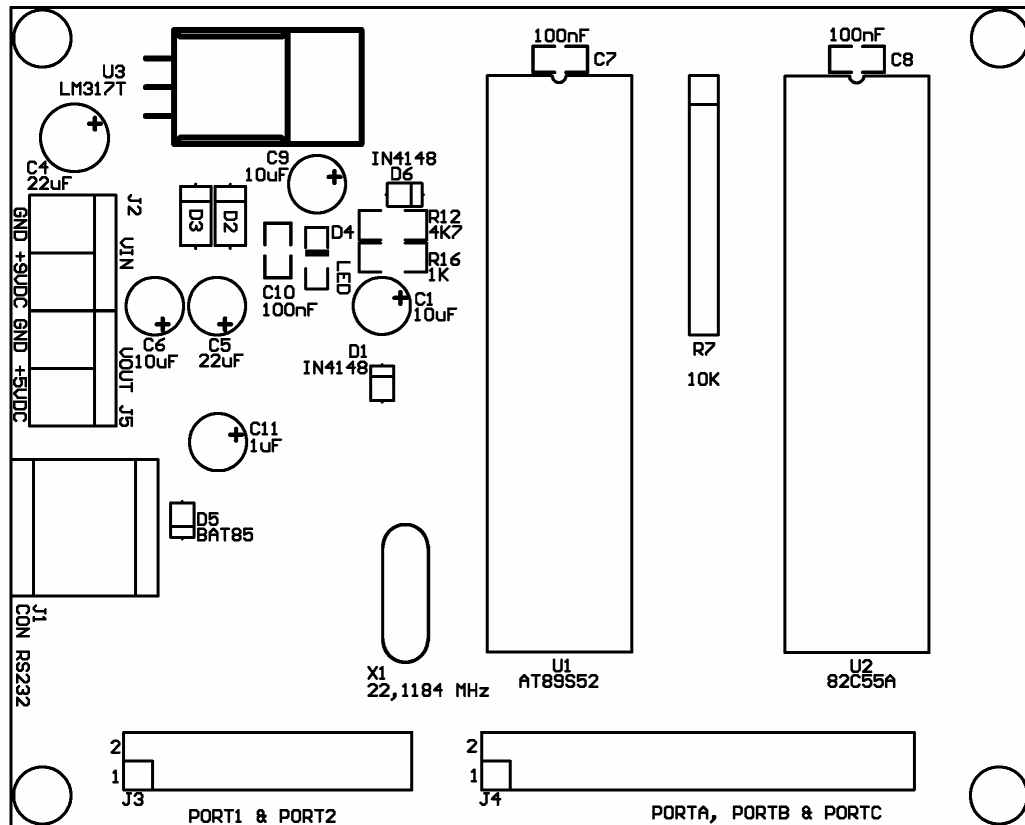
- PC AT Pentium® IBM Compatible dengan Serial Port (COM1 / COM2).
- CD-ROM Drive.
- Ruang hard disk minimum 2 Mbytes.

Perangkat Lunak:

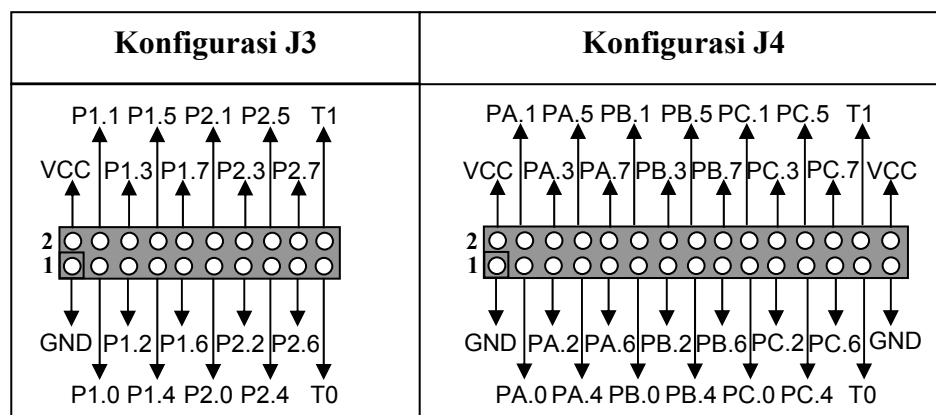
- Sistem Operasi MS-DOS®, PC-DOS™, atau Windows® 9x ke atas.
- Borland® Delphi 5.0 atau Turbo® Pascal 7.0.
- File-file dalam CD:  
SERPPI.EXE, PASPPI.EXE, SERLIB.DLL, SERPAS.TPU, MANUAL SERIAL PPI.PDF, QUICK START SERIAL PPI.PDF, 89S52.PDF, 82C55.PDF

## 2. PERANGKAT KERAS SERIAL PPI

### 2.1. TATA LETAK KOMPONEN SERIAL PPI



### 2.2. ALOKASI DAN SPESIFIKASI PORT



Spesifikasi untuk Port 1, 2, Counter 0, dan Counter 1 adalah sebagai berikut:

Simbol	Parameter	Nilai	Satuan
$I_{OL}$	Arus saat output berlogika '0'	1,6	mA
$I_{OH}$	Arus saat output berlogika '1'	-10	$\mu$ A

$I_{OL}$  maksimum per pin adalah 10 mA.

$I_{OL}$  maksimum per port adalah 15 mA.

$I_{OL}$  maksimum untuk semua port adalah 71 mA.

Spesifikasi untuk Port A, B, dan C adalah sebagai berikut:

Simbol	Parameter	Nilai	Satuan
$I_{OL}$	Arus saat output berlogika '0'	2,5	mA
$I_{OH}$	Arus saat output berlogika '1'	-100	$\mu$ A

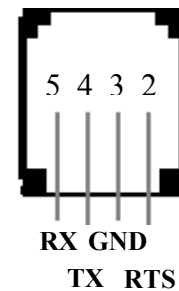
Keterangan lebih lanjut dapat dilihat pada datasheet IC yang bersangkutan.

### 2.3. HUBUNGAN SERIAL PPI DENGAN KOMPUTER

SERIAL PPI dapat dihubungkan dengan COM port komputer atau dengan kontroler lain yang juga memiliki interface UART RS-232. Perhatikan hubungan jalur komunikasinya.

COM Port Komputer DB9 Male	SERIAL PPI RJ11 Female
RTS (Pin 7)	RTS (Pin 2)
GND (Pin 5)	GND (Pin 3)
TX (Pin 3)	TX (Pin 4)
RX (Pin 2)	RX (Pin 5)

J1 Tampak Depan



### 2.4. MENCOBA SERIAL PPI DENGAN SERPPI.EXE

- Hubungkan kabel serial antara SERIAL PPI dan komputer.
- Hubungkan perangkat output (misalnya rangkaian LED) ke port SERIAL PPI.
- Hubungkan sumber tegangan. Hubungkan juga referensi Ground antara rangkaian tambahan dengan Ground pada SERIAL PPI.
- Jalankan program SERPPI.EXE (under Windows<sup>®</sup>), pilih COM yang digunakan dan tekan tombol **Start**. LED yang dihubungkan pada port SERIAL PPI akan menyala bergantian.

### 2.5. MENCOBA SERIAL PPI DENGAN PASPPI.EXE

- Hubungkan kabel serial antara SERIAL PPI dan komputer.
- Hubungkan perangkat output (misalnya rangkaian LED) ke port SERIAL PPI.
- Hubungkan sumber tegangan. Hubungkan juga referensi Ground antara rangkaian tambahan dengan Ground pada SERIAL PPI.
- Jalankan program PASPPI.EXE (under DOS). Ketik COM port yang digunakan dan tekan Enter. LED yang dihubungkan pada port SERIAL PPI akan menyala bergantian.

## 3. PERANGKAT LUNAK SERIAL PPI

Waktu yang dibutuhkan SERIAL PPI mulai menyala hingga siap dioperasikan (Start-up Time) = 25 ms.

### 3.1. SPESIFIKASI UART RS-232

Secara default, komunikasi UART RS-232 bekerja pada **Baud Rate 9600 bps, 8 Data Bit, No Parity Bit, 1 Stop Bit, No Flow Control**. Pilihan baud rate yang lain terdapat pada **bagian 3.2.4**.

### 3.2. COMMAND

Semua transaksi data selalu dimulai dengan mengirimkan Command ke SERIAL PPI.

#### Command

<b>M2</b>	<b>M1</b>	<b>M0</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
-----------	-----------	-----------	----------	----------	----------	----------	----------

#### Mode

<b>M2</b>	<b>M1</b>	<b>M0</b>	<b>MODE</b>
0	0	0	<i>Tidak Terpakai</i>
0	0	1	Byte Transfer
0	1	0	Bit Set/Reset
0	1	1	Counter
1	0	0	Baud Rate
1	1	X	<i>Tidak Terpakai</i>

#### 3.2.1. BYTE TRANSFER

#### Command

<b>0</b>	<b>0</b>	<b>1</b>	<b>P2</b>	<b>P1</b>	<b>A1</b>	<b>A0</b>	<b>R/W</b>
----------	----------	----------	-----------	-----------	-----------	-----------	------------

#### Mode

Command untuk Byte Transfer akan mengakses salah satu dari Port 1, Port 2, Port A, Port B, Port C, atau Control Word. Mengakses lebih dari satu port pada saat yang sama tidak diperkenankan.

Proses tulis/output dan proses baca/input dibedakan oleh R/W. R/W diberi nilai 1 untuk proses pembacaan dari port dan diberi nilai 0 untuk proses penulisan ke port.

Proses baca akan diikuti oleh satu byte hasil pembacaan port yang dikirim oleh modul SERIAL PPI. Proses tulis harus diikuti oleh satu byte yang akan dikirimkan ke port.

<b>Mode</b>	<b>P2</b>	<b>P1</b>	<b>A1</b>	<b>A0</b>	<b>R/W</b>	<b>Proses</b>
001	0	0	0	0	0	Tulis ke Port A
	0	0	0	0	1	Baca dari Port A
	0	0	0	1	0	Tulis ke Port B
	0	0	0	1	1	Baca dari Port B
	0	0	1	0	0	Tulis ke Port C
	0	0	1	0	1	Baca dari Port C
	0	0	1	1	0	Tulis ke Control Word
	0	0	1	1	1	<i>Tidak Terpakai</i>
	0	1	0	0	0	Tulis ke Port 1

Mode	P2	P1	A1	A0	R/W	Proses
001	0	1	0	0	1	Baca dari Port 1
	0	1	0	1	X	Tidak Terpakai
	0	1	1	X	X	Tidak Terpakai
	1	0	0	0	0	Tulis ke Port 2
	1	0	0	0	1	Baca dari Port 2
	1	0	0	1	X	Tidak Terpakai
	1	0	1	X	X	Tidak Terpakai
	1	1	X	X	X	Tidak Terpakai

Secara default, kondisi semua port pada saat awal adalah:

- Port 1 dan Port 2 : sebagai input, bernilai FFH
- Port A, Port B, Port C : Tidak terprogram sebagai input maupun output

Untuk memprogram Port A, B, dan C, proses inisialisasi harus dilakukan dengan mengirim 1 byte ke Control Word sebelum ketiga port digunakan.

MSB (Most Significant Bit)	LSB (Least Significant Bit)						
D7	D6	D5	D4	D3	D2	D1	D0

Simbol	Fungsi
D7	Set Flag, diberi nilai 1 untuk mengaktifkan PPI Port.
D6 & D5	Mode Select untuk Port A dan C Upper (bit 7 – bit 4), bernilai 00 untuk mode 0, 01 untuk mode 1, dan 10 atau 11 untuk mode 2. Umumnya diberi nilai 00.
D4	Port A, diberi nilai 1 untuk input dan diberi nilai 0 untuk output.
D3	Port C Upper, diberi nilai 1 untuk input dan diberi nilai 0 untuk output.
D2	Mode Select untuk Port B dan C Lower (bit 3 – bit 0), diberi nilai 0 untuk mode 0 dan diberi nilai 1 untuk mode 1. Umumnya diberi nilai 0.
D1	Port B, diberi nilai 1 untuk input dan diberi nilai 0 untuk output.
D0	Port C Lower, diberi nilai 1 untuk input dan diberi nilai 0 untuk output.

Untuk penggunaan mode lain, lihat keterangan lebih lengkap pada datasheet 82C55.

Selain diakses per byte, PPI Port dapat juga diakses per bit (Bit Set/Reset). Akses per bit hanya terbatas pada Port C saja dan hanya sebagai output saja. Meski menggunakan Port C, tetapi akses per bit bukan berarti mengakses langsung ke alamat Port C. Akses per bit dapat dilakukan dengan mengirim 1 byte ke Control Word.

MSB	LSB						
D7	D6	D5	D4	D3	D2	D1	D0

Simbol	Fungsi
D7	Set Flag, diberi nilai 0 untuk mengaktifkan akses per bit.
D6, D5, D4	Bit bersifat don't care (dapat diberi nilai 0 atau 1). Disarankan untuk mengisi dengan nilai 0.
D3, D2, D1	Bit Select untuk menentukan bit pada Port C yang akan diakses. Keterangan lebih jelas ditunjukkan oleh tabel 2.
D0	Bit Set/Reset untuk Port C. Diberi nilai 1 untuk Set (memberi logika 1 pada bit) dan 0 untuk Reset (memberi logika 0 pada bit).

	Bit dari Port C							
	7	6	5	4	3	2	1	0
<b>D3</b>	1	1	1	1	0	0	0	0
<b>D2</b>	1	1	0	0	1	1	0	0
<b>D1</b>	1	0	1	0	1	0	1	0

### 3.2.2. BIT SET/RESET

#### Command

0	1	0	P2/ $\overline{P1}$	D2	D1	D0	V
---	---	---	---------------------	----	----	----	---

#### Mode

Command untuk Bit Set/Reset akan menulis/output salah satu bit dari Port 1 atau Port 2.

Pemilihan bit ditentukan oleh nilai D0, D1, dan D2. Sedangkan nilai yang dituliskan ke bit ditentukan oleh V.

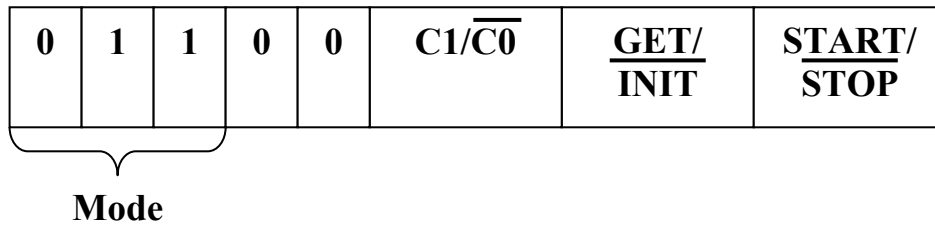
D2	D1	D0	Bit yang Diakses
0	0	0	Bit 0
0	0	1	Bit 1
0	1	0	Bit 2
0	1	1	Bit 3
1	0	0	Bit 4
1	0	1	Bit 5
1	1	0	Bit 6
1	1	1	Bit 7

Mode	P2/ $\overline{P1}$	D2	D1	D0	V	Proses
010	0	X	X	X	0	Clear/Reset (memberi logika 0) pada Bit Port 1
	0	X	X	X	1	Set (memberi logika 1) pada Bit Port 1
	1	X	X	X	0	Clear/Reset (memberi logika 0) pada Bit Port 2
	1	X	X	X	1	Set (memberi logika 1) pada Bit Port 2



### 3.2.3. COUNTER

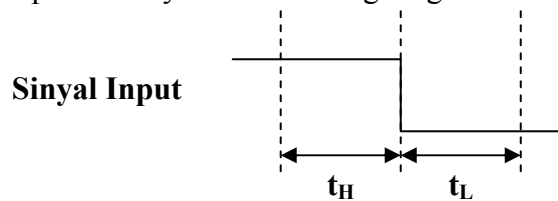
#### Command



Command untuk Counter akan mengakses salah satu dari Counter 0 atau Counter 1.

**Start** dan **Stop** berfungsi untuk mengaktifkan atau menghentikan counter dan hanya bekerja apabila **Init** diberi nilai 0.

Register Counter 0 dan Counter 1 akan bertambah jika pin yang bersangkutan (T0 atau T1) mendapatkan sinyal sesuai timing diagram berikut.



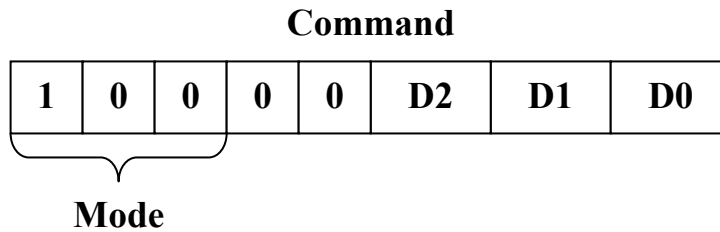
Symbol	Keterangan	Nilai	Satuan
$t_H$	Durasi sinyal berlogika '1' sebelum transisi	0,55	$\mu s$
$t_L$	Durasi sinyal berlogika '0' setelah transisi	0,55	$\mu s$

Pembacaan dengan Get akan diikuti oleh dua byte hasil pembacaan nilai counter yang dikirim oleh modul SERIAL PPI. Data pertama adalah nilai Least Significant Byte, sedangkan data kedua adalah nilai Most Significant Byte.

Mode	00	C1/ $\overline{C0}$	GET/ $\overline{INIT}$	START/ $\overline{STOP}$	Proses
011	00	0	0	0	Menghentikan Counter 0
	00	0	0	1	Mengaktifkan Counter 0
	00	0	1	0	Get Counter 0 Data
	00	0	1	1	<i>Tidak Terpakai</i>
	00	1	0	0	Menghentikan Counter 1
	00	1	0	1	Mengaktifkan Counter 1
	00	1	1	0	Get Counter 1 Data
	00	1	1	1	<i>Tidak Terpakai</i>
	01	X	X	X	<i>Tidak Terpakai</i>
	1X	X	X	X	<i>Tidak Terpakai</i>

Secara default, kondisi awal Counter 0 dan Counter 1 adalah tidak aktif.

### 3.2.4. BAUD RATE



Command untuk Baud Rate akan mengaktifkan salah satu dari beberapa setting Baud Rate.

Pemilihan setting baud rate ditentukan oleh nilai D0, D1, dan D2.

Mode	00	D2	D1	D0	Baud Rate
100	00	0	0	0	9600 bps
	00	0	0	1	19200 bps
	00	0	1	0	38400 bps
	00	0	1	1	57600 bps
	00	1	0	0	115200 bps
	01	1	X	1	<i>Tidak Terpakai</i>
	1X	X	X	X	<i>Tidak Terpakai</i>

Secara default, kondisi baud rate pada saat awal adalah:

- Baud Rate : 9600 bps

### 3.3. RUTIN DLL DAN TPU

SERIAL PPI memiliki library berupa file SERLIB.DLL yang dapat digunakan oleh bahasa pemrograman yang dapat mengakses file tersebut (misalnya Borland<sup>®</sup> Delphi, Borland<sup>®</sup> C++, dll). Selain itu SERIAL PPI juga memiliki unit berupa file SERPAS.TPU yang dapat digunakan oleh bahasa pemrograman Turbo Pascal. Kedua file ini dapat digunakan untuk mempermudah user dalam pemrograman.

Berikut ini adalah rutin-rutin yang digunakan dalam SERLIB.DLL dan SERPAS.TPU:

#### **IOFlag**

Fungsi : Memeriksa status komunikasi terakhir.

Tipe : Function

Input : -

Output : IOFlag Tipe : Boolean

Keterangan : -

Metode : Panggil rutin ini untuk memeriksa apakah SERIAL PPI membalas dengan kode Acknowledge (FAH) pada komunikasi terakhir.

Jika command terakhir dibalas dengan FAH oleh SERIAL PPI, maka IOFlag bernilai = True.

Jika command terakhir tidak dibalas dengan FAH oleh SERIAL PPI, IOFlag bernilai = False.











```

begin
    r1.text:=inttohex(PortRead('1'),2);
    if IOFlag then edit1.text:='Success' else
    edit1.text:='Error';
end;

//Aktifkan Counter 0
procedure TForm1.StartC0Click(Sender: TObject);
begin
    if not CountStatus(0,True) then showmessage('Error');
    if IOFlag then edit1.text:='Success' else
    edit1.text:='Error';
end;

//Baca register Counter 0
procedure TForm1.GetC0Click(Sender: TObject);
begin
    g0.text:=inttohex(CountRead(0),4);
    if IOFlag then edit1.text:='Success' else
    edit1.text:='Error';
end;

//Hentikan Counter 0
procedure TForm1.StopC0Click(Sender: TObject);
begin
    if not CountStatus(0,False) then showmessage('Error');
    if IOFlag then edit1.text:='Success' else
    edit1.text:='Error';
end;

```

Berikut ini adalah cuplikan program dalam Turbo<sup>©</sup> Pascal 7.0 untuk mengakses Port 2.7 per bit, mengubah baud rate, serta melakukan reset.

```

{mengubah bit P2.7 secara toggle}
if p27 then
begin
    if BitWrite(2,nl,false) then
    begin
        p27:=false;
        textbackground(4);
        clrscr;
        end;
    end
else
begin
    if BitWrite(2,nl,true) then
    begin
        p27:=true;
        textbackground(2);
        clrscr;
        end;
    end;

{mengubah baud rate SERIAL PPI}
if ym=2 then DeviceBaud(19200)
{mengubah baud rate komputer}
else if ym=3 then ComBaud(19200)
{melakukan reset}
else if ym=4 then DeviceReset;

```



### 3.5. KERANGKA PROGRAM

Bagi user yang ingin membuat program aplikasi SERIAL PPI dengan menggunakan rutin yang sudah ada, maka file SERLIB.DLL atau SERPAS.TPU harus digunakan/dipanggil.

*SERLIB.DLL* merupakan library yang akan selalu digunakan untuk setiap aplikasi SERIAL PPI yang menggunakan pemrograman under Windows®.

*SERPAS.TPU* merupakan unit yang akan selalu digunakan untuk setiap aplikasi SERIAL PPI yang menggunakan pemrograman Pascal (under DOS).

Kerangka pemrograman SERIAL PPI menggunakan Borland® Delphi 5.0 adalah sebagai berikut :

```
unit SERIALPPI;

interface

uses
  Windows, Forms;           { Isi sesuai kebutuhan }

type
  TForm1 = class(TForm)
    .           { Deklarasi komponen/prosedur/fungsi user }
    .
    .
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;           { Deklarasi variabel program user }
  .
  .
  .
implementation

{$R *.DFM}

.           { Nama rutin SerLib.dll }
stdcall; external 'SerLib.dll';

.           { Panggil rutin SerLib.dll sesuai kebutuhan }
.
.

.           { Prosedur/fungsi program user }
.
.
end.
```

Kerangka pemrograman SERIAL PPI menggunakan Turbo® Pascal 7.0 adalah sebagai berikut:

```
program SERIALPPI;

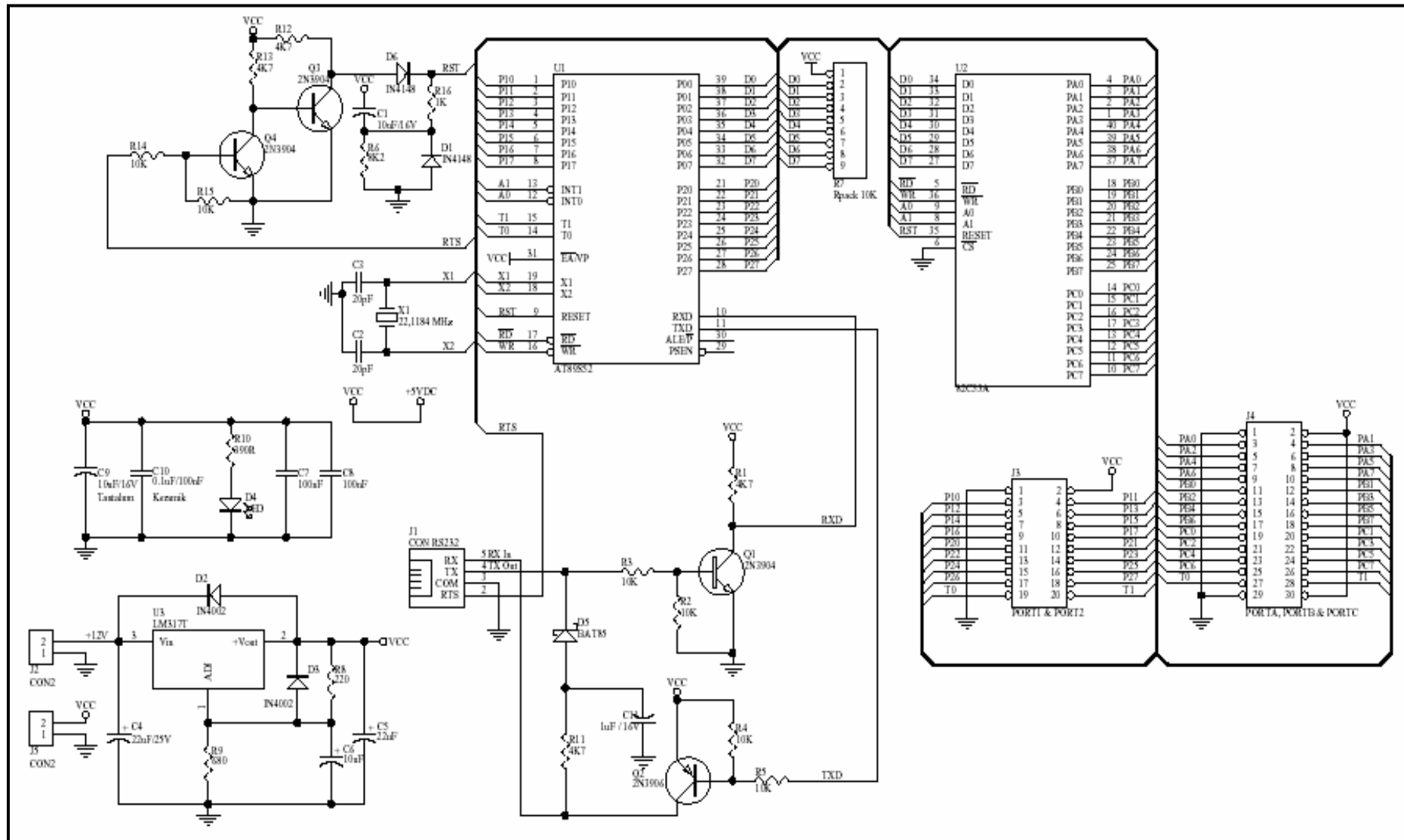
uses
  Dos, Crt, Serpas;           { Isi sesuai kebutuhan }
```





# LAMPIRAN A

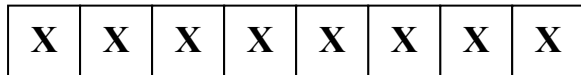
## SKEMA SERIAL PPI



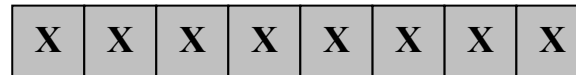
## LAMPIRAN B

### PROTOKOL SERIAL PPI

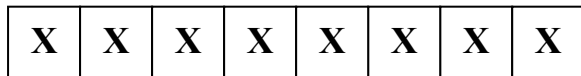
Byte Dikirim oleh Master



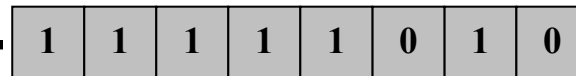
Byte Dikirim oleh Device



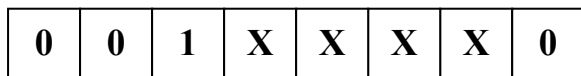
Command yang Dikenali



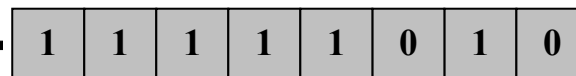
Acknowledge



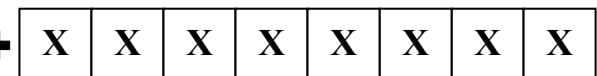
Write Byte



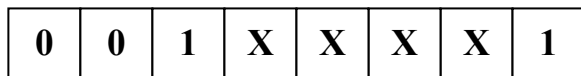
Acknowledge



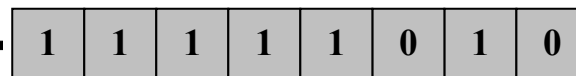
Data



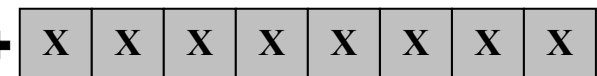
Read Byte



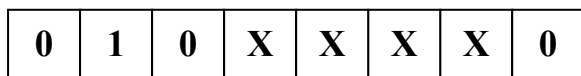
Acknowledge



Data



Clear Bit



Acknowledge

