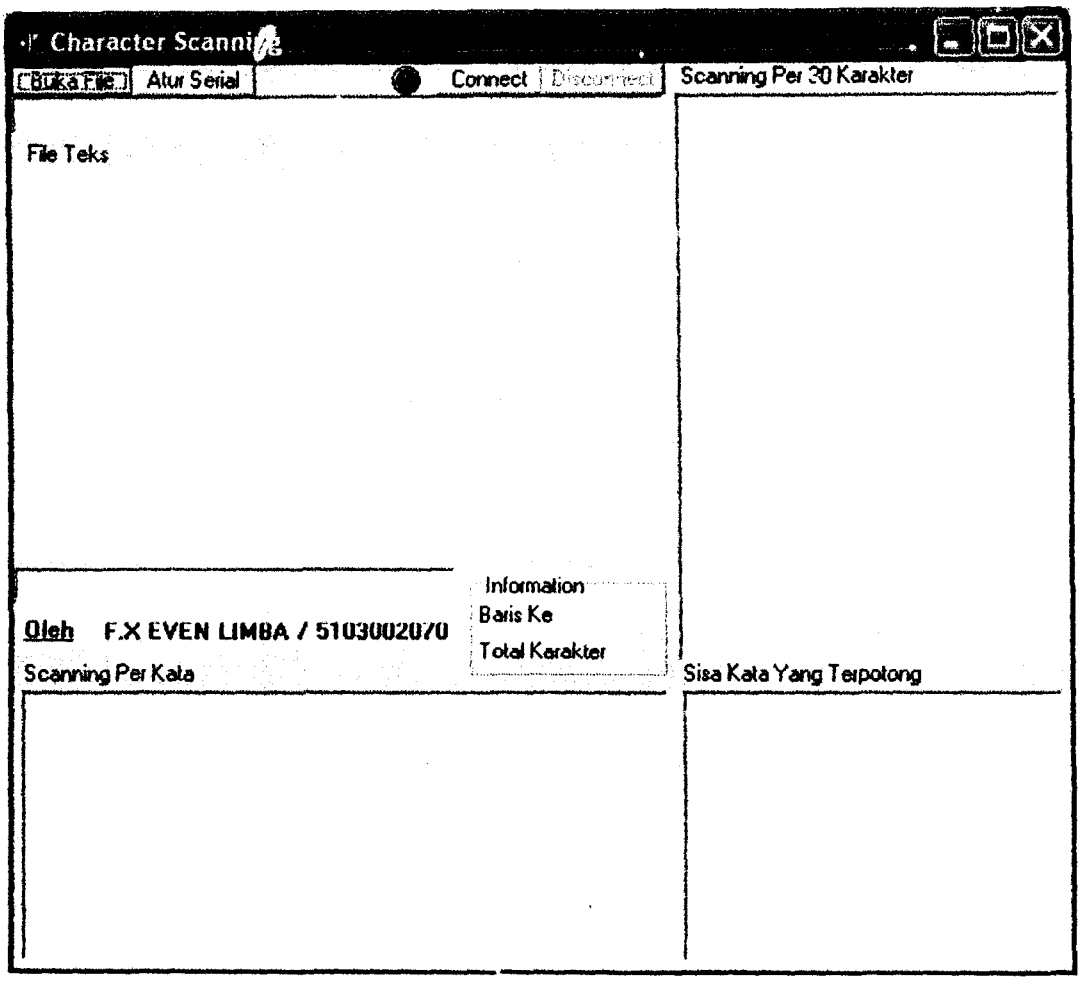
LAMPIRAN A

LANGKAH – LANGKAH PENGGUNAAN PROGRAM *"CHARACTER SCANNING"*

LANGKAH-LANGKAH PENGGUNAAN PROGRAM *"CHARACTER SCANNING"*

Berikut ini adalah langkah – langkah penggunaan menggunakan program "*Character Scanning*" : (Khusus no 1 dan 2 dioperasikan oleh orang normal dengan kata lain memerlukan bantuan orang normal)

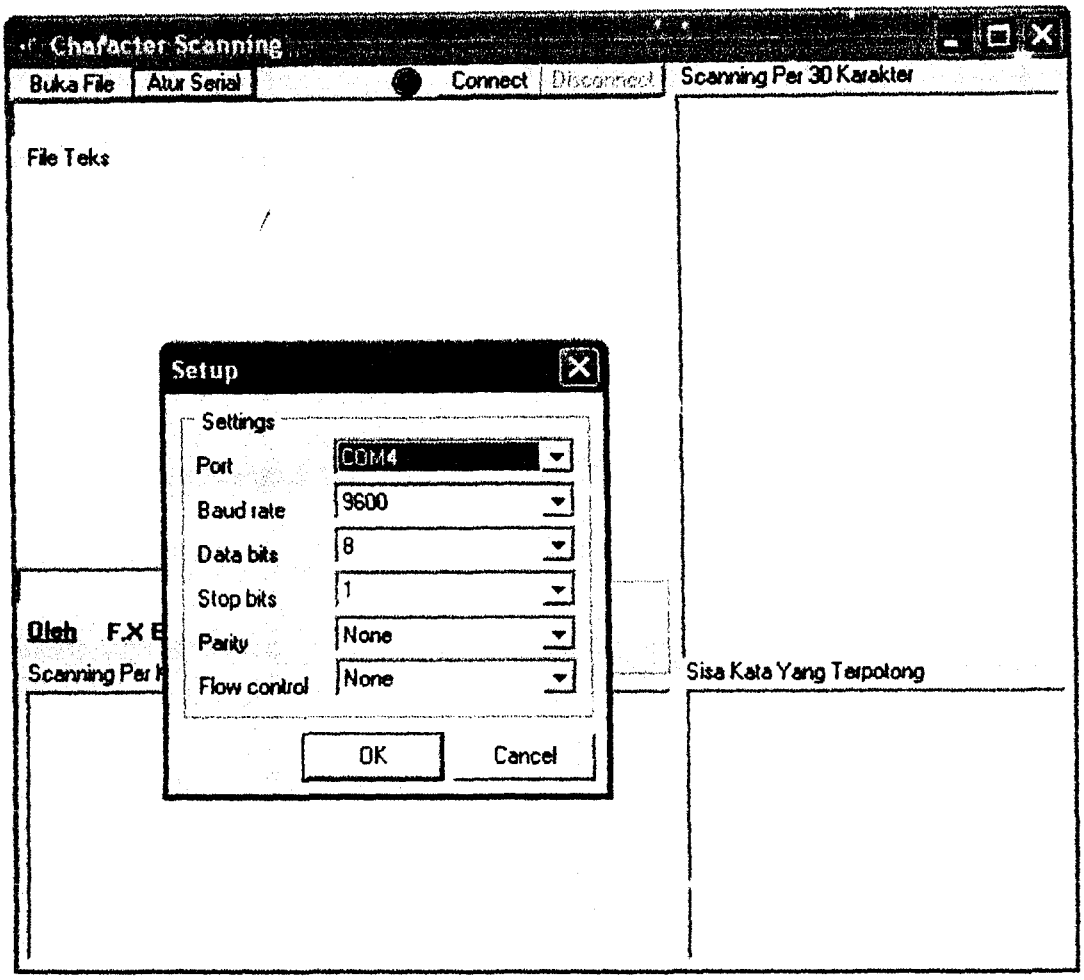
1. Buka program "*Character Scanning*".



Gambar L.1. Tampilan Awal Program "*Character Scanning*"

Keterangan lebih jelas tentang fonn ini dapat dilihat pada gambar 3.14. – sub-bab 3.3.1.

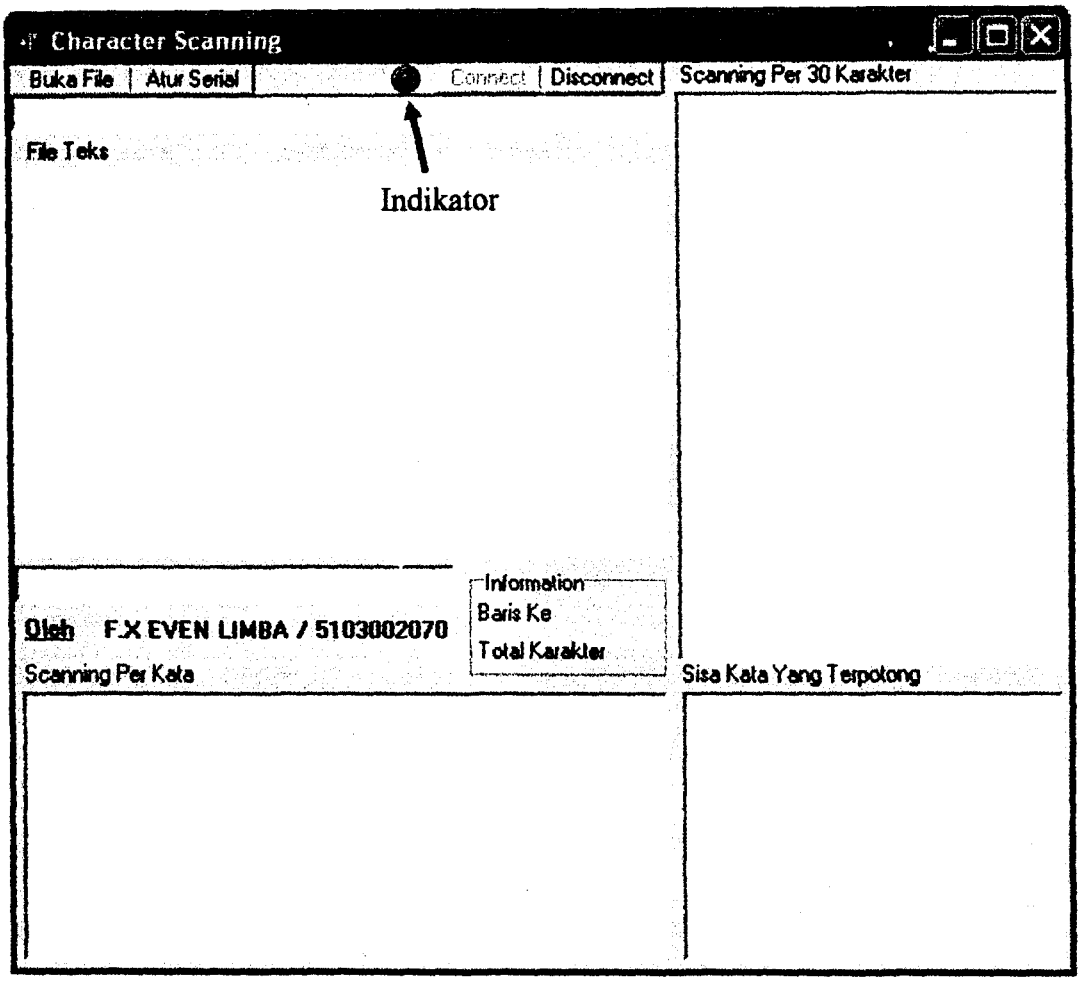
2. Melakukan Pengaturan *Serial Port*



Gambar L.2. Pengaturan *Serial Port*

Pada pengaturan ini nilai awal telah diatur menurut "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC". Namun juga memungkinkan untuk memilih *Port* konfigurasi komunikasi serial yang ingin dipakai untuk "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC". (Selain *port* konfigurasi lain telah dicocokkan dengan mikrokontroler, jadi tidak perlu diubah (statis atau telah ditentukan)).

3. Menghubungkan PC dengan "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC"

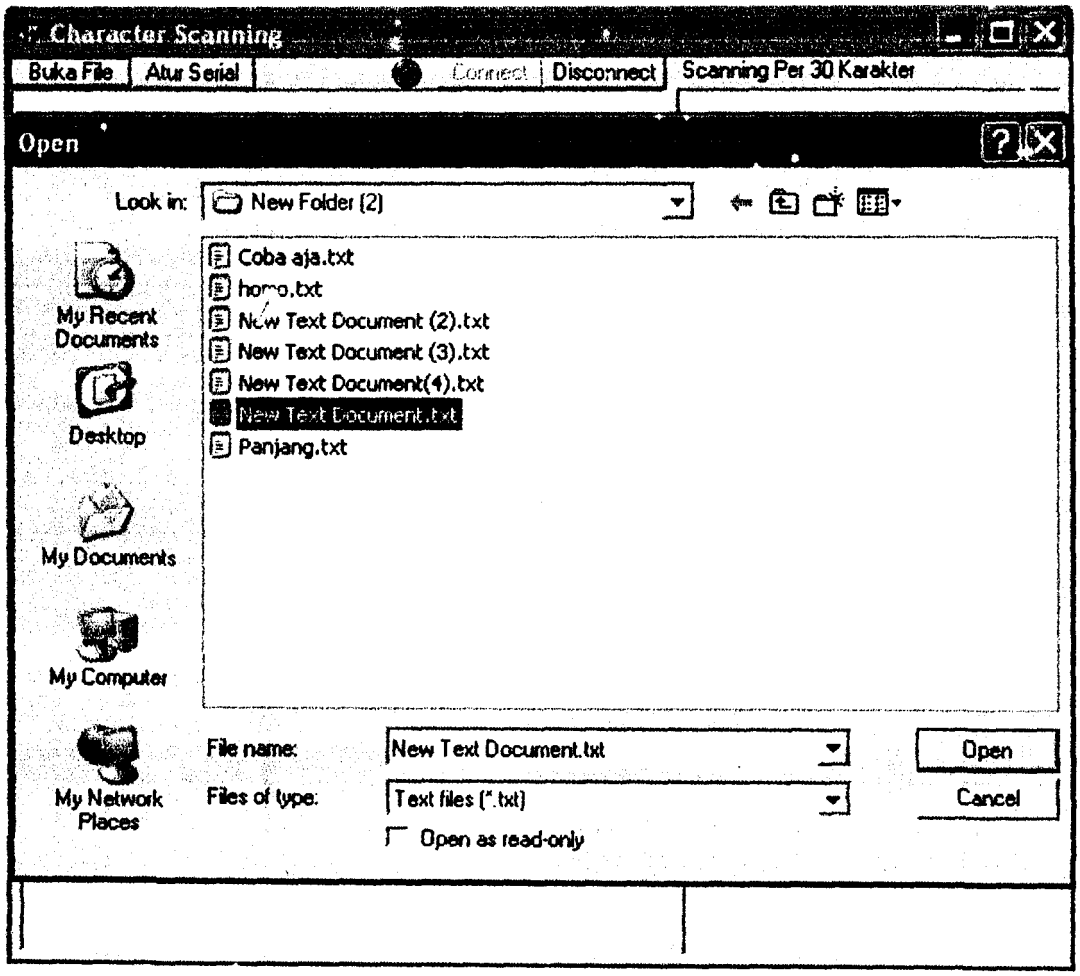


Gambar L.3. Menghubungkan PC dan "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC"

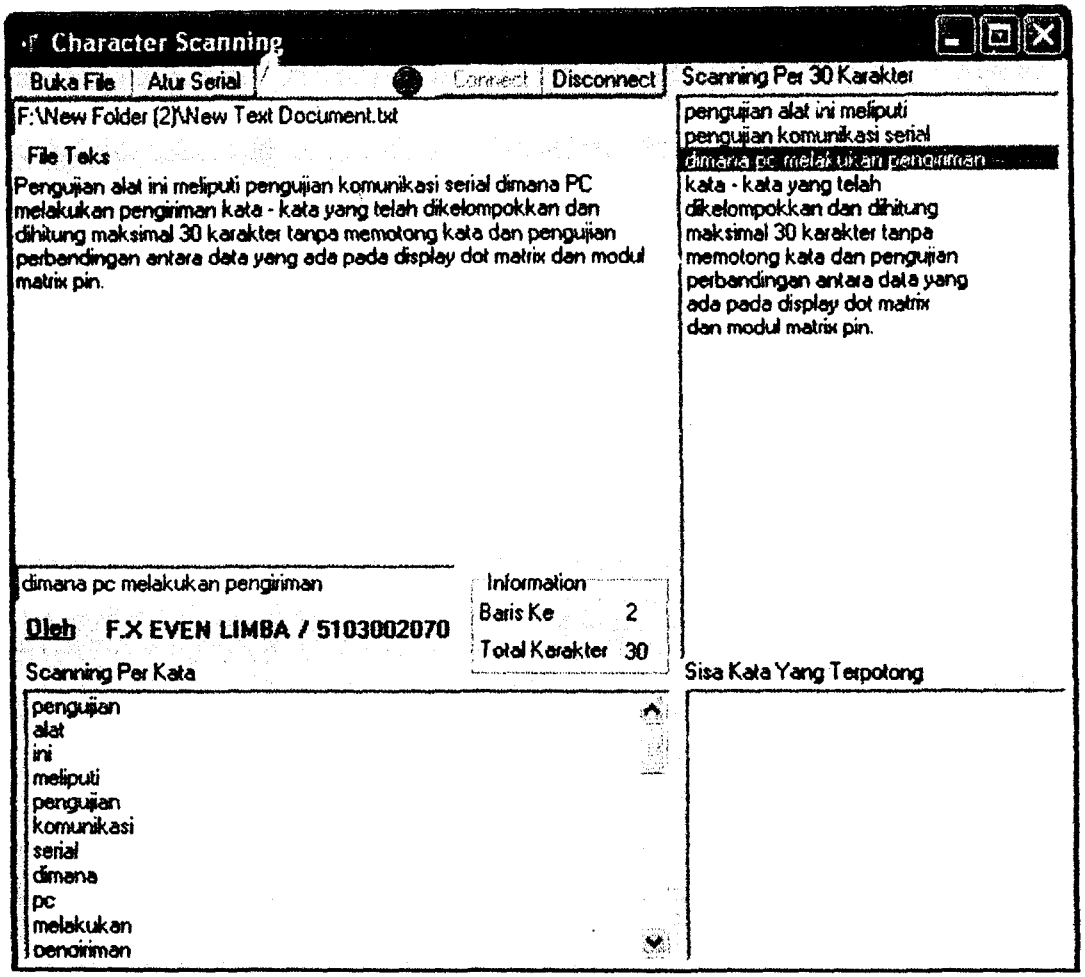
Klik tombol "*connect*" untuk membuka koneksi port serial. Pastikan "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC" terhubung dengan port serial pada PC. Port serial siap dipakai ditandai dengan indikator lampu merah disamping tombol "*connect*".

4. Buka File Teks

Klik tombol "*Open Teks*" dan menu "*Browser*" akan tampil, pilih file teks yang ingin dibuka dan klik "*open*". Untuk lebih jelasnya dapat dilihat pada gambar L.4 dan gambar L.5.



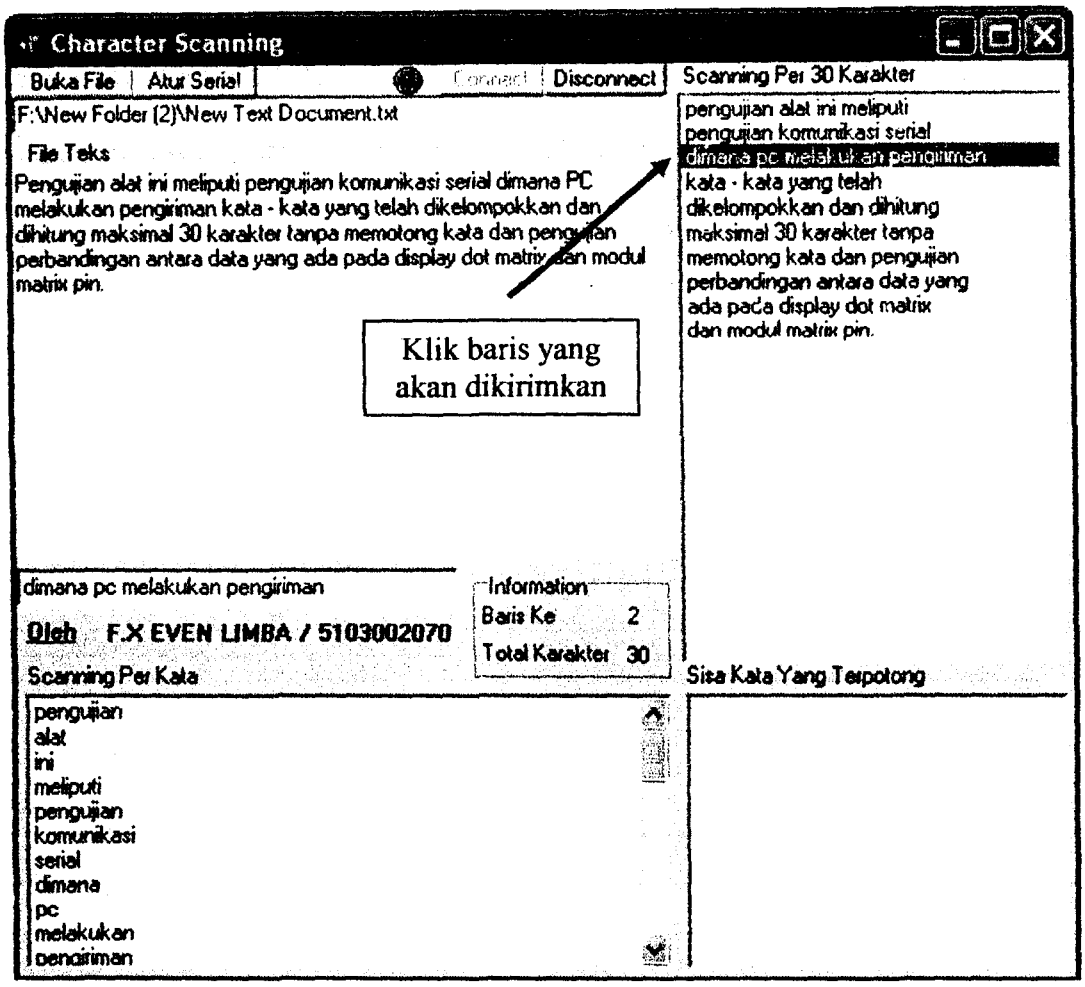
Gambar L.4. Menu "*Browser*" pada "*Open Teks*"



Gambar L.5. Hasil Setelah Membuka File Teks

Setelah meng-klik tombol "*open*" maka teks file yang telah dipilih secara otomatis akan dibuka dan *scanning* karakter akan segera dilakukan. Untuk mengenali spasi program akan melakukan seleksi karakter jika ditemukan spasi maka satu kata tersebut akan disimpan. Hasil setelah membuka file teks dapat dilihat pada gambar L.5.

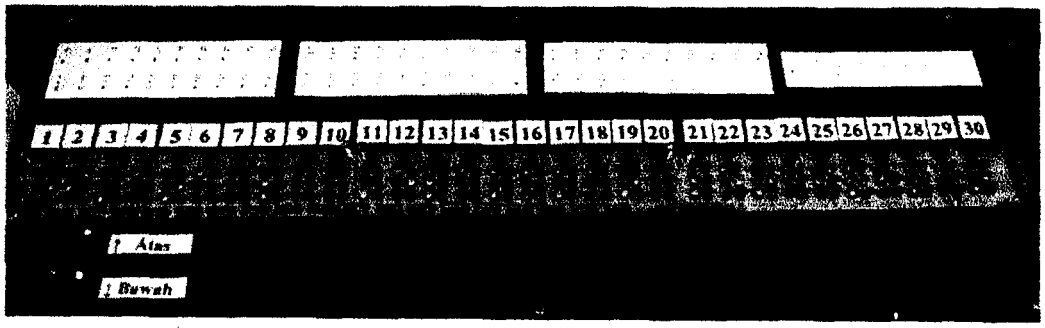
5. Melakukan Pengiriman *scanning* 30 karakter



Gambar L.6. Pengiriman *Scanning* 30 karakter

Klik baris yang ingin dikirimkan pada kolom *scanning* 30 karakter (harus dilakukan pada pertama kali sistem dinyalakan). Untuk selanjutnya navigasi akan dilakukan melalui tombol atas dan tombol bawah pada "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC". Hasil dari pengiriman *scanning* 30 karakter dapat dilihat pada gambar L.7.

Ketika sistem sudah berjalan, jika ingin mengakses file teks baru perlu mengatur ulang agar "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC" kembali pada baris ke 0 dengan menekan tombol atas sampai baris berada pada posisi baris ke 0. Hal ini dilakukan untuk melakukan *update* pada data yang ada pada "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC".

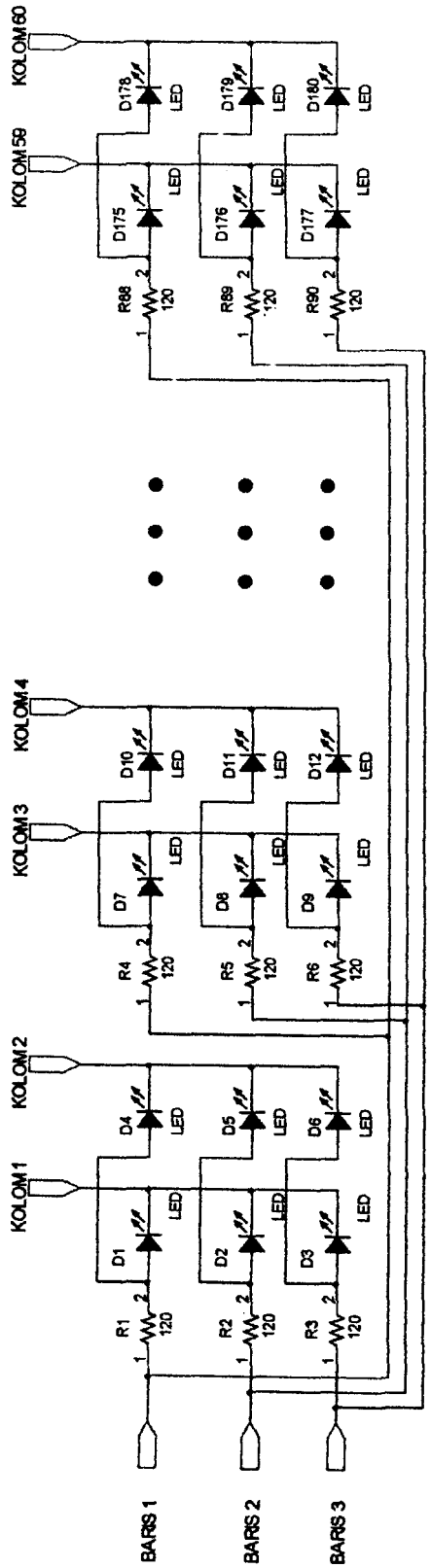


Gambar L.7. Hasil Pengiriman *Scanning 30* Karakter

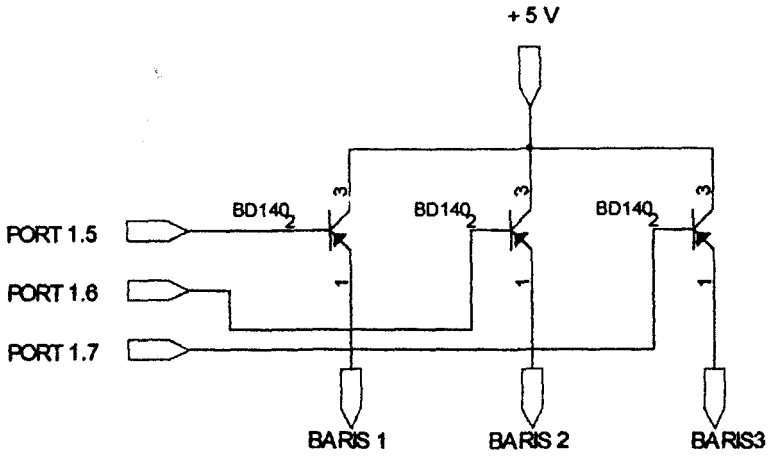
Navigasi pengiriman *scanning* 30 karakter melalui tombol atas dan tombol bawah pada "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC" akan secara otomatis memilih indeks sebelumnya dari indeks yang sedang dikirimkan bila menekan tombol atas, dan memilih indeks sesudahnya bila menekan tombol bawah.

LAMPIRAN B

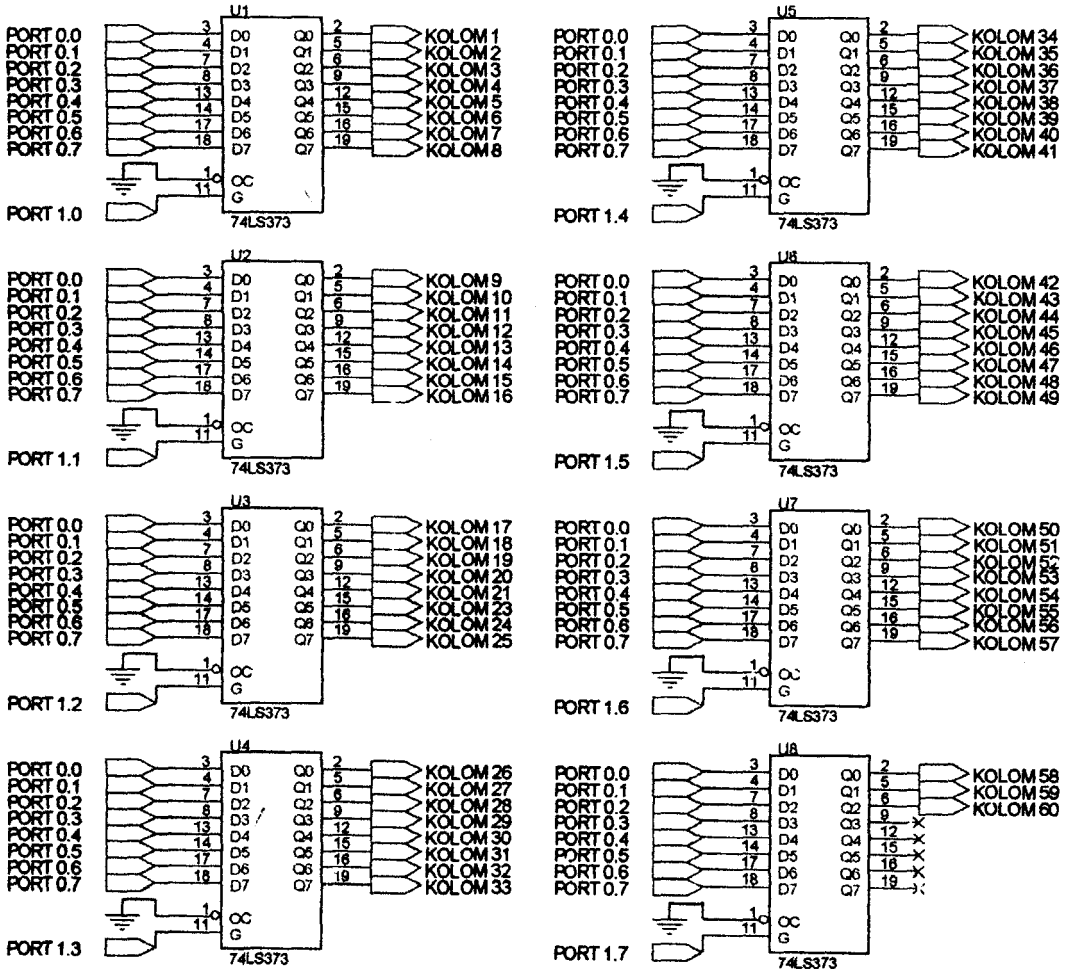
GAMBAR RANGKAIAN PERANGKAT KERAS



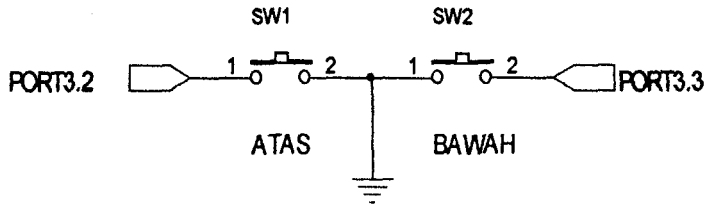
Gambar L.8. Rangkaian *Display* Dot Matrix



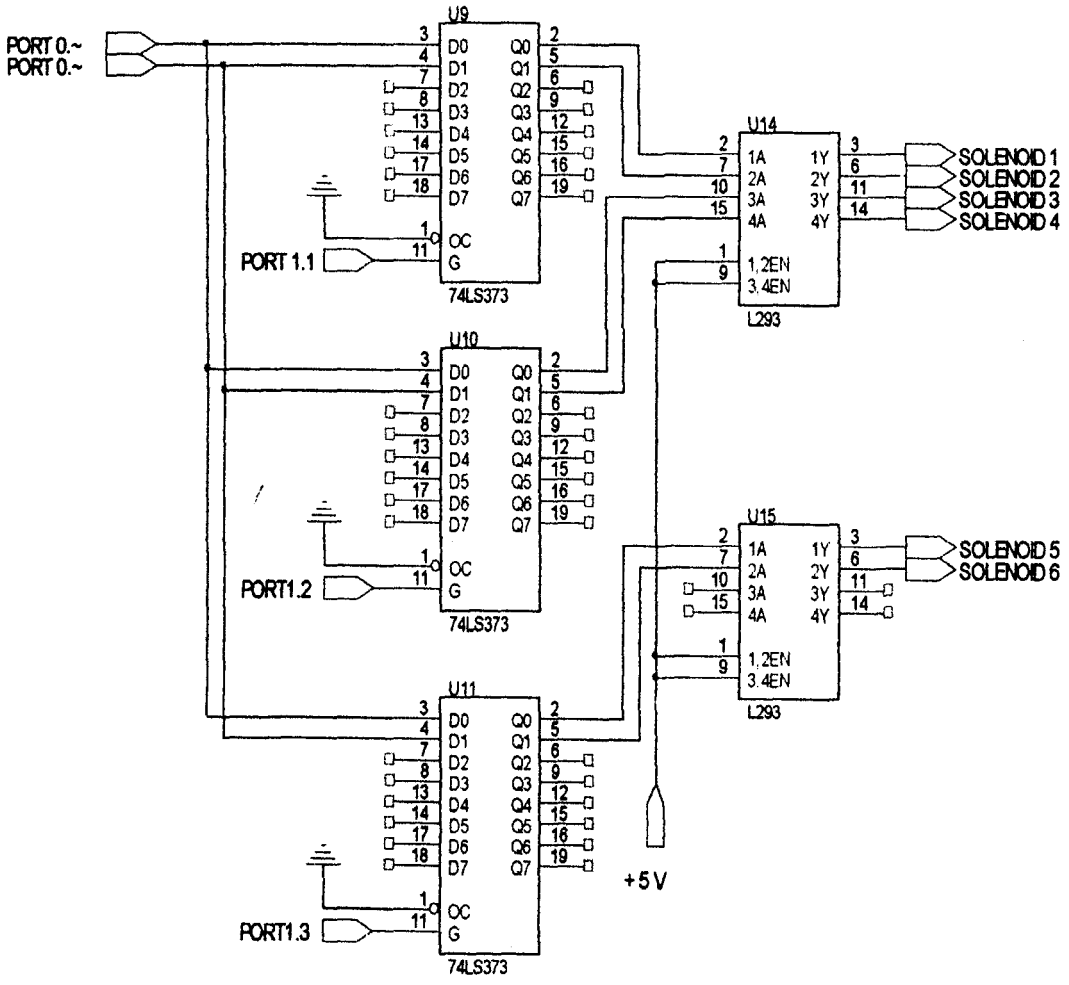
Gambar L.9. Rangkaian *Driver Display* Dot Matrix



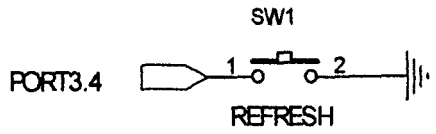
Gambar L.10. Rangkaian *Latch Display* Dot Matrix



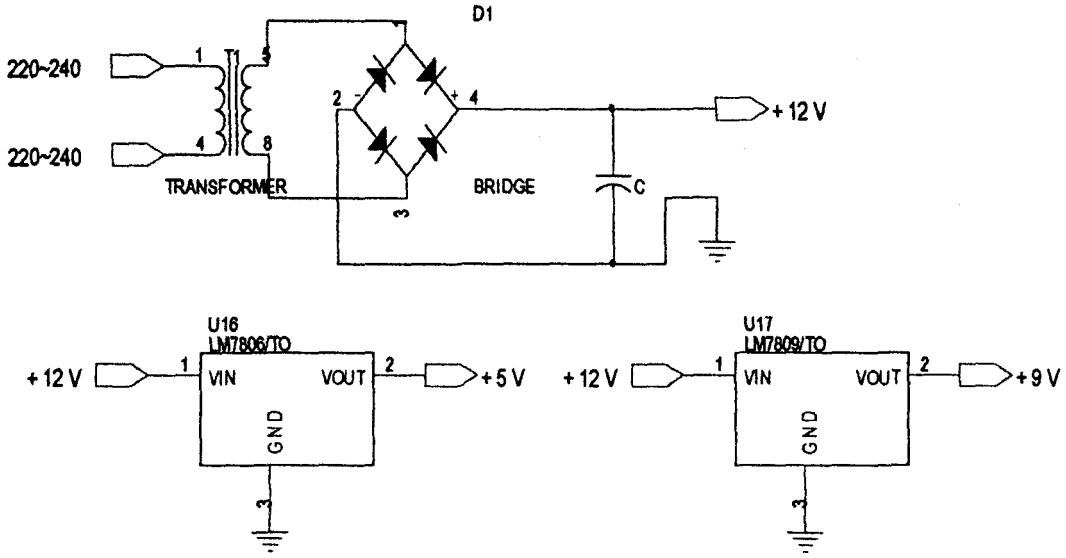
Gambar L.11. Rangkaian Tombol Atas dan Tombol Bawah



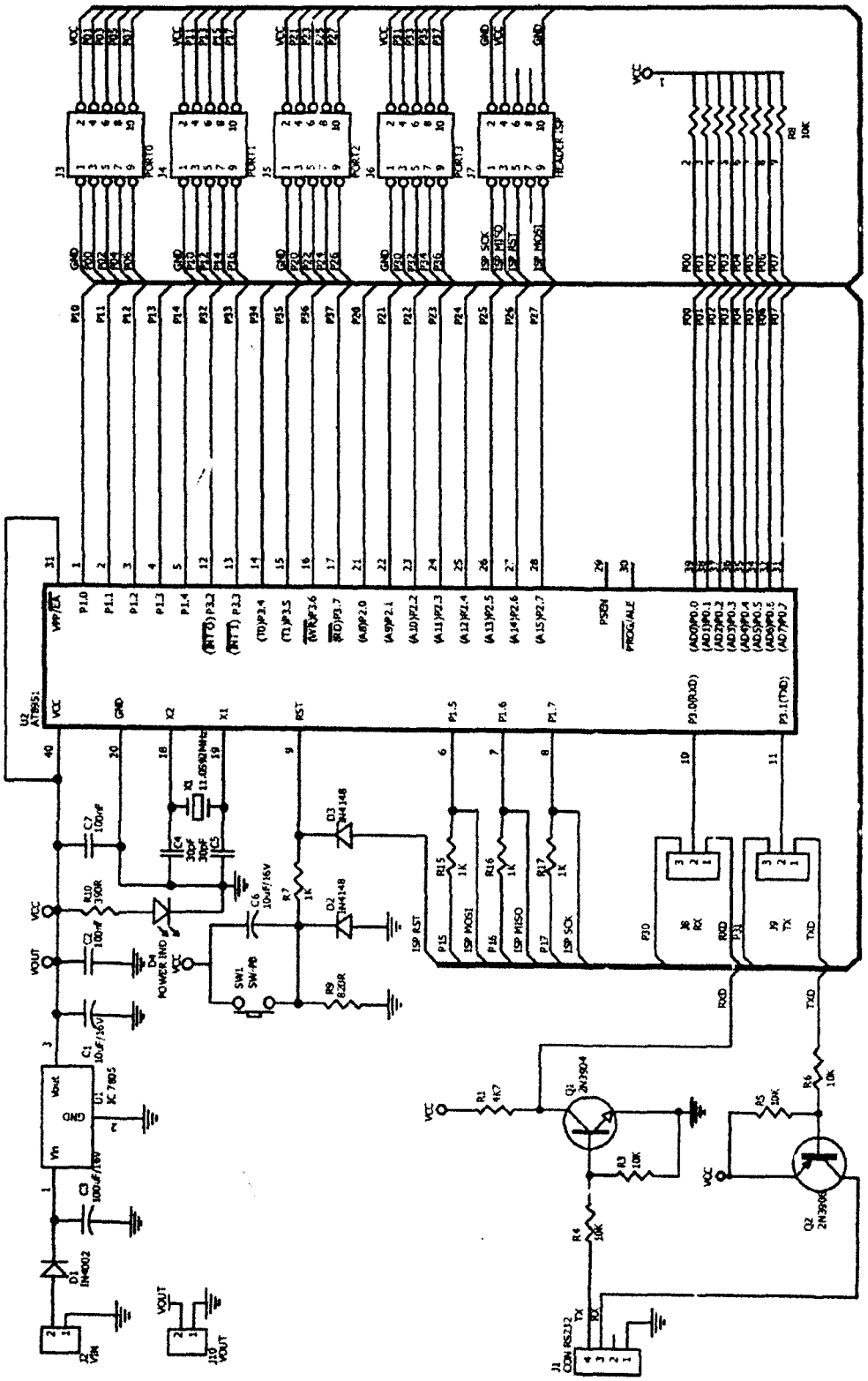
Gambar L.12. Rangkaian *Latch* Modul Matrix Pin



Gambar L.13. Tombol *Refresh* pada Modul Matrix Pin



Gambar L.14. Rangkaian Catu Daya



Gambar L.15. Rangkaian Modul DT-51

LAMPIRAN C

PERANGKAT LUNAK

- Program "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC"

```
;############################## KETERANGAN ##############################
;DIBAWAH INI MERUPAKAN INIALISASI ALAMAT UNTUK MEMPERMUDAH PEMROGRAMAN
;########################################################################
 baris        equ 70h
 baris1       equ 71h
 kolom        equ 72h
 kolom1       equ 73h
 buff         equ 74h
 buff1        equ 75h
 data1        equ 76h
 data2        equ 77h
 kunci        equ 78h
 kunci1       equ 79h
 ganti        equ 7ah
 ganti1       equ 7bh
 hit1         equ 7ch
 hit2         equ 7dh
 flag         equ 7eh
 bank_flag    equ 7fh
 bank2        equ 6fh
 ;########################################################################
ORG    0000H
       JMP    INIT

org    0003h
       jmp eks0

org    000bh
       jmp    interupt_timer0

org    0013h
       jmp eks1

ORG    0023H
       jmp    terima

org 350h
;########################################################################
;PROSEDUR INTERUPSI SERIAL
;########################################################################

 terima:
       JNB    RI, $                ;Tunggu sampai data masuk
       MOV    A, SBUF              ;Masukkan hex yg diterima ke
       mov    @r0,A
              INC R0
       CLR    RI                   ;Tutup penerimaan
       CLR    ES                   ;Tutup Serial interrupt
       SETB   ES
       RETI
;########################################################################
```

```
;#############################################################################
;PROSEDUR INTERUPSI EXTERNAL 0 (TOMBOL ATAS)
;#############################################################################
 eks0:
       clr    ex0
       call reset
       call   delay_5ms
       jnb    p3.2,$

       mov    p3,#0FFh
       pop   dph
       pop    dpl
       mov    dph,#00h
       mov    dpl,#00h

       mov    a,#61h
       mov    sbuf,a
       jnb    ti,$
       clr    ti
       mov    a,#00h

       push   dpl
       push   dph
       setb   ex0
       reti
;#############################################################################

;#############################################################################
;PROSEDUR INTERUPSI EXTERNAL 1 (TOMBOL BAWAH)
;#############################################################################
 eks1:
       clr    ex1
              call reset
       call   delay_5ms
       jnb    p3.3,$

       mov    p3,#0FFh
       pop   dph
       pop    dpl
       mov    dph, #00h
       mov    dpl, #00h

       mov    a,#62h
       mov    sbuf,a
       jnb    ti,$
       clr    ti
       mov    a,#00h

       push   dpl
       push   dph
       setb   ex1
       reti
;#############################################################################
```

```
;###########################################################################
;PROSEDUR RESET
;###########################################################################
 reset:
       MOV    R0,#04EH
       MOV    A,#00H
       mov    baris,#00h
       mov    baris1,#00h
       mov    kolom,#00h
       mov    kolom1,#00h
       mov    kunci,#01h
       mov    kunci1,#01h
       mov    hit1,#00h
       mov    hit2,#00h
       mov    ganti,#0C0h
       mov    ganti1,#0E1h
       mov    p0,#0ffh
       mov    p1,#0ffh
       mov    p2,#0ffh
       mov    p3,#0ffh
       MOV    6CH,#41H
       MOV    6DH,#41H
       ret

;###########################################################################

;###########################################################################
;PROGRAM UTAMA
;###########################################################################
org 400h
  INIT:
       mov    bank_flag,#00h
       mov    flag,#00h
       mov    sp,#07h
       MOV    SCON, #50H
       MOV    TMOD, #22H
       mov    th0,#0
       MOV    TL1, #0FDH
       MOV    TH1, #0FDH
       setb   TR1
       MOV    PCON, #00H
       MOV    IE, #97H
       clr    tr0
       call reset
       jmp    LOOP

LOOP:
       MOV    A,R0
       CJNE   A,#06CH,LOOP
       clr es
       JMP START1
;###########################################################################
```

```
;###########################################################################
;PROSEDUR ISI LATCH SOLENOID
;MODUL MATRIX PIN
;UPDATE MULAI DARI AMBIL KARAKTER, KUNCI KARAKTER....
;###########################################################################
ORG 500H
start3:
       call   reset
       call   delay


start1:

       mov    R1,#4EH        ;BACA TULISAN
       mov    A,kolom1
       ADD    A,R1
       MOV    R1,A
       MOV    A,@R1
       subb   a,#20h


       mov    dptr/#huruf    ;MEngeluarkan data huruf ke PA
       MOV    B,#3
       MUL    AB             ;kali isi a dengan reg b digunakan untuk enuju
                             ke huruf yg ditunjukkan

       CLR    C
       ADD    A,DPL          ;menambahkan isi acc dengan 8bit  hasil
                             disimpan di acc
       MOV    DPL,A
       MOV    A,B
       ADDC   A,DPH
       MOV    DPH,A

       mov    a,baris1
       movc   a,@a+dptr

       mov    buff1,a
       mov    a,hit2
       cjne   a,#00h,next11
       mov    data2,buff1

 next11:
       cjne   a,#01h,next21
       mov    a,buff1
       rr     a
       rr     a
       add    a,data2
       mov    data2,a
       mov    a,hit2
 next21:
       cjne   a,#02h,next31
       mov    a,buff1
       swap   a
       add    a,data2
       mov    data2,a
       mov    a,hit2
 next31:
       cjne   a,#03h,next41
       mov    a,buff1
       swap   a
       rr     a
       rr     a
```

```
      add    a,data2
      mov    data2,a
      mov    a,hit2
next41:
      inc kolom1
      inc hit2
      mov    a,hit2
      cjne   a,#04h,start1
      mov    p0,data2
      mov    hit2,#00h

      mov    p2,kunci1
      call   delay_2
      mov    p2,#00H

      mov    a,kunci1
      rl            a
      mov    kunci1,a

      mov    a,kunci1

      MOV    A,KOLOM1
      CJNE   A,#020H,teruskanlagi
      jmp    terusin1
      teruskanlagi:
      mov    a,kunci1
      CJNE   A,#01h,start1
 terusin1:

      mov    p1,ganti1
      aCALL  DELAY_2
      mov    p1,#0e0h

      mov    a,ganti1
      cjne   a,#0E1h,lanjut13
      mov    ganti1,#0E2h
      jmp    lanjut3

 lanjut13:
      cjne   a,#0E2h,lanjut23
      mov    ganti1,#0E4h
      jmp    lanjut3

 lanjut23:
      cjne   a,#0E4h,lanjut3
      mov    ganti1,#0E1h
      jmp lanjut3

 lanjut3:
      mov    kolom1,#00h
      mov    kunci1,#01h
      inc    baris1
      mov    a,baris1
      cjne   a,#03h,jump1
      mov    baris1,#00h
      jmp  start

 jump1:
      jmp    start1

;#############################################################################
```

```
;#############################################################################
;PROSEDUR SCANNING LED (DISPLAY DOT MATRIX)
;MELAKUKAN AMBIL KARAKTER, KUNCI KARAKTER DAN CEK MODUL MATRIX PIN
;#############################################################################
org 600h
 start:
       setb   tr0
       mov    a,flag
       CJNE   a,#01h,scan_lagi
       jmp    start3
 scan_lagi:
       mov    R1,#4EH        ;BACA TULISAN
       mov    A,kolom
       ADD    A,R1
       MOV    R1,A
       MOV    A,@R1
       subb   a,#20h


       mov    dptr,#huruf    ;MEngeluarkan data huruf ke PA
       MOV    B,#3
       MUL    AB             ;kali isi a dengan reg b digunakan untuk menuju
                             ke huruf yg ditunjukkan
       CLR    C                                  ;
       ADD    A,DPL          ;menambahkan isi acc dengan 8bit  hasil
                             disimpan di acc
       MOV    DPL,A            ;
       MOV    A,B              ;
       ADDC   A,DPH            ;
       MOV    DPH,A            ;

       mov    a,baris          ;
       movc   a,@a+dptr        ;

       mov    buff,a

       mov    a,hit1

       cjne   a,#00h,next1
       mov    data1,buff
 next1:
       cjne   a,#01h,next2
       mov    a,buff
       rr     a
       rr     a
       add    a,data1
       mov    data1,a
       mov    a,hit1
              next2:
              cjne   a,#02h,next3
                     mov    a,buff
                     swap   a
                     add    a,data1
                     mov    data1,a
                     mov    a,hit1
              next3:
              cjne   a,#03h,next4
                     mov    a,buff
                     swap   a
                     rr     a
                     rr     a
                     add    a,data1
                     mov    data1,a
```

```
        mov    a,hit1
next4:
inc kolom
inc hit1
mov    a,hit1
cjne   a,#04h,start
mov    p0,data1
mov    hit1,#00h

mov    p2,kunci
call   delay_2
mov    p2,#00H

mov    a,kunci
rl             a
mov    kunci,a

mov    a,kunci


MOV    A,KOLOM
CJNE   A,#020H,START
jmp terusin
lagilagi:

mov    a,kunci
CJNE   A,#01h,scan_lagi2
jmp    terusin
scan_lagi2:
jmp    start

terusin:
mov    p1,ganti
aCALL  DELAY_5MS
mov    p1,#0f0h

mov    a,ganti
cjne   a,#0C0h,lanjut1
mov    ganti,#0A0h
jmp    lanjut

lanjut1:
cjne   a,#0A0h,lanjut2
mov    ganti,#060h
jmp    lanjut

lanjut2:
cjne   a,#060h,lanjut
mov    ganti,#0C0h
jmp lanjut

lanjut:
mov    kolom,#00h
mov    kunci,#01h
inc    baris
mov    a,baris
cjne   a,#03h,jump
mov    baris,#00h
 jmp start

jump:
jmp   start
;###############################################################################
```

```
;###########################################################################
;--PROSEDUR-DELAY-------------------------------------------------------
;###########################################################################
ORG 450H
       delay_5ms:
       mov r7,#100
       d_1 : mov r6,#10
             djnz r6,$
             djnz r7,d_1
       ret

delay:
       mov r7,#255
d : mov r6,#255
       djnz r6,$
       djnz r7,d
       ret

       delay_2:
       mov r7,#1
       d_1b : mov r6,#5
             djnz r6,$
             djnz r7,d_1b
       ret
;###########################################################################

;###########################################################################
;PROSEDUR INTERUPSI TIMER
;UNTUK CEK MODUL MATRIX PIN TERPASANG
;###########################################################################
       interupt_timer0:
       mov    bank2,a
       jb            p3.4,gakada
       inc    bank_flag
       mov    a,bank_flag
       cjne   a,#100,cek_2
       jmp    tampil
       cek_2:
       cjne   a,#102,selesai
       mov    flag,#00h
       mov    bank_flag,#101
       jmp selesai
       gakada:
       mov    bank_flag,#00h
   mov flag,#00h
   jmp selesai
       tampil:
       mov    flag,#01h
       selesai:
       mov    a,bank2
       reti
;###########################################################################
```

```
;##################################################################
;DATABASE UNTUK KODE ASCII
;KODE ASCII AKAN MENGAMBIL REFERENSI DISINI UNTUK MENDAPATKAN KODE
;BRAILLE
;DIBAWAH INI DATABASE YANG BERISIKAN KARAKTER BRAILLE
;##################################################################
org 800h
huruf:
        db 0C0h,0C0h,0C0h,      ;{SP}0    20h         0
        db 080h,040h,000h,      ;{!} 8    21h       1
        db 0C0h,0C0h,0C0h,      ;{"*} 16            2
        db 080h,080h,000h,      ;{#} 24             3
        db 000h,040h,080h,      ;{$}32                    4
        db 000h,0C0h,080h,      ;{%}40              5
        db 000h,040h,000h,      ;{&} 48              6
        db 0C0h,0C0h,0C0h,      ;{'*}56              7
        db 040h,000h,000h,      ;{(}64              8
        db 080h,000h,000h,      ;{)}72              9
         db 040h,0C0h,080h,     ;{*} 80              10
         db 080h,0C0h,000h,     ;{+} 88              11
         db 0C0h,0C0h,080h,     ;{,} 96              12
         db 0C0h,0C0h,000h,     ;{-}                 13
         db 080h,0C0h,080h,     ;{.}                 14
         db 080h,0C0h,040h,     ;{/}                 15
         db 0C0h,080h,000h,     ;{0}           30h        16
         db 0C0h,040h,0C0h,     ;{1}                 17
         db 0C0h,040h,040h,     ;{2}                 18
         db 0C0h,000h,0C0h,     ;{3}                 19
         db 0C0h,000h,080h,     ;{4}                 20
         db 0C0h,040h,080h,     ;{5}                 21
         db 0C0h,000h,040h,     ;{6}                 22
        db 0C0h,000h,000h,      ;{7}                 23
         db 0C0h,040h,000h,     ;{8}                 24
         db 0C0h,080h,040h,     ;{9}                 25
         db 040h,080h,080h,     ;{:}                 26
         db 0C0h,080h,080h,     ;{;}                 27
         db 040h,040h,080h,     ;{<}                 28
         db 000h,000h,000h,     ;{=}                 29
         db 080h,080h,040h,     ;{>}                 30
         db 000h,080h,080h,     ;{?}                 31
         db 080h,0C0h,0C0h,     ;{@}                 32
         db 040h,0C0h,0C0h,     ;{A}                                  33=21H
         db 040h,040h,0C0h,     ;{B}                 34
         db 000h,0C0h,0C0h,     ;{C}                 35
         db 000h,080h,0C0h,     ;{D}                 36
         db 040h,080h,0C0h,     ;{E}                 37
         db 000h,040h,0C0h,     ;{F}                 38
         db 000h,000h,0C0h,     ;{G}                 39
         db 040h,000h,0C0h,     ;{H}                 40
         db 080h,040h,0C0h,     ;{I}                           41
         db 080h,000h,0C0h,     ;{J}
         db 040h,0C0h,040h,     ;{K}
         db 040h,040h,040h,     ;{L}
         db 000h,0C0h,040h,     ;{M}
         db 000h,080h,040h,     ;{N}
         db 040h,080h,040h,     ;{O}
         db 000h,040h,040h,     ;{P}
         db 000h,000h,040h,     ;{Q}
         db 040h,000h,040h,     ;{R}
         db 080h,040h,040h,     ;{S}
         db 080h,000h,040h,     ;{T}
         db 040h,0C0h,000h,     ;{U}
         db 040h,040h,000h,     ;{V}
```

```
        db 080h,000h,080h,      ;{W}
        db 000h,0C0h,000h,      ;{X}
        db 000h,080h,000h,      ;{Y}
        db 040h,080h,000h,      ;{Z}
        db 080h,040h,080h,      ;{[}
        db 040h,000h,080h,      ;{\}
        db 000h,000h,080h,      ;{]}
        db 080h,080h,0C0h,      ;{^};PAKMAN2
        db 080h,080h,000h,      ;{_}
        db 0C0H,0C0H,0C0H,      ;{**}
        db 040h,0C0h,0C0h,      ;{a}
        db 040h,040h,0C0h,      ;{b}
        db 000h,0C0h,0C0h,      ;{c}
        db 000h,080h,0C0h,      ;{d}
        db 040h,080h,0C0h,      ;{e}
        db 000h,040h,0C0h,      ;{f}
        db 000h,000h,0C0h,      ;{g}
        db 040h,000h,0C0h,      ;{h}
        db 080h,040h,0C0h,      ;{i}
        db 080h,000h,0C0h,      ;{j}
        db 040h,0C0h,040h,      ;{k}
        db 040h,040h,040h,      ;{l}
        db 000h,0C0h,040h,      ;{m}
        db 000h,080h,040h,      ;{n}
        db 040h,080h,040h,      ;{o}
        db 000h,040h,040h,      ;{p}
        db 000h,000h,040h,      ;{q}
        db 040h,000h,040h,      ;{r}
        db 080h,040h,040h,      ;{s}
        db 080h,000h,040h,      ;{t}
        db 040h,0C0h,000h,      ;{u}
        db 040h,040h,000h,      ;{v}
        db 080h,000h,080h,      ;{w}
        db 000h,0C0h,000h,      ;{x}
        db 000h,080h,000h,      ;{y}
        db 040h,080h,000h,      ;{z}
        db 080h,040h,080h,      ;{[}
        db 040h,000h,080h,      ;{|}
        db 000h,000h,080h,      ;{]}
        db   0c0h,0c0h,0c0h,    ;{~}
        db   0c0h,0c0h,0c0h,    ;{del}
end
```

- Program "Character Scanning"

```
//#############################################################################
//INIALISASI DELPHI
//#############################################################################
unit Teks;

interface

uses
  Windows, Messages, strutils,SysUtils, Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, OleServer, Word2000, Menus, CPortCtl,
CPort,
  ExtCtrls;

type
  TForm1 = class(TForm)
    Button1: TButton;
    OpenDialog1: TOpenDialog;
    RichEdit1: TRichEdit;
    Edit1: TEdit;
    ListBox1: TListBox;
    Button2: TButton;
    Edit2: TEdit;
    FileTeks: TLabel;
    ListBox2: TListBox;
    Label1: TLabel;
    Label2: TLabel;
    Button5: TButton;
    Button6: TButton;
    ComLed1: TComLed;
    GroupBox1: TGroupBox;
    Label5: TLabel;
    Label3: TLabel;
    Label6: TLabel;
    Timer1: TTimer;
    ComPort1: TComPort;
    Label7: TLabel;
    Label8: TLabel;
    Label4: TLabel;
    Edit3: TEdit;
    ListBox3: TListBox;
    Label9: TLabel;
    Edit4: TEdit;
    ListBox4: TListBox;
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  //  procedure Button3Click(Sender: TObject);
    procedure ListBox2Click(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
    procedure ComPort1RxChar(Sender: TObject; Count: Integer);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
    procedure ListBox4Click(Sender: TObject);

  private
    { Private declarations }
```

```
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  tampung : array[1..1000]of string;
  tampung30 : array[1..1000] of string;
implementation

{$R *.dfm}

//#############################################################################
//PROSEDUR TOMBOL BUKA FILE
//#############################################################################

procedure TForm1.Button1Click(Sender: TObject);
var
  I,a,b,c,d,k,e,g,j,toi,po,o: integer;
  ab:longint;
  F: TextFile;
  FirstLine,baca,teks,buf :string;
  hitung30,sisa: string[100];

begin
        listbox1.Clear;
        listbox2.Clear;
        listbox3.Clear;
        listbox4.Clear;
        edit2.Text:='';
        edit3.Text:='';
        edit4.text:='';
  for  ab:=1 to 1000 do
        begin
        tampung[ab]:='';
        tampung30[ab]:='';
        end;
  richedit1.Text := '';
  edit1.Text :='';
  OpenDialog1.Options := [ofAllowMultiSelect, ofFileMustExist];
  OpenDialog1.Filter := 'Text files (*.txt)|*.txt';
  OpenDialog1.FilterIndex := 1; { Penunjuk filter }
  if OpenDialog1.Execute then
    with OpenDialog1.Files do
      for I := 0 to Count - 1 do
      begin
        AssignFile(F, Strings[I]);  { next file in Files property }
        edit1.Text := strings[I];   {tampilkan Path}
        reset(F);
        while not eof(F) do
         begin
          Readln(F, FirstLine);  { Read the first line out of the file }
          RichEdit1.Lines.Append(FirstLine); { Add the line to the memo }
          // masukke listbox3
          if firstline <> '' then
          begin
          listbox3.Items.Add(firstline);
          end;
           end;
        CloseFile(F);
      end;
```

```
// masukkan ke editteks
begin
  edit3.Text:='';
  edit3.Text:=listbox3.Items.Strings[0];
  for a:=1 to listbox3.Count-1 do
  begin
  edit3.text:=edit3.Text+' '+listbox3.Items.Strings[a];
  end;
  for g:=1 to length(edit3.Text) do
  begin
   if midstr(edit3.Text,g,1) <> #09 then
   edit4.Text:=edit4.Text+midstr(edit3.Text,g,1);
  end;
end;

//kelompokkan kata
  baca:=lowercase(edit4.text);
  j:=1;
   for c:=1 to length(baca) do
begin

        if midstr(baca,c,1)<>' ' then
        begin
              tampung[j]:=tampung[j]+midstr(baca,c,1);
        end
        else
         begin
          if midstr(baca,c+1,1)<>' ' then
            begin
                j:=j+1;
            end;
         end;
end;

if midstr(tampung[1],1,1) ='' then
begin
for k:=2 to j do
begin
        if tampung[k]<>'' then
        begin
        listbox1.Items.add((tampung[k])+' ');
        end;
end;
end;

if midstr(tampung[1],1,1) <>'' then
begin
 for k:=1 to j do
begin
        if tampung[k]<>'' then
        begin
        listbox1.Items.add((tampung[k])+' ');
        end;
end;
end;
// kelompokkan 30 karakter
begin
g:=0;
 b:=0;
c:=listbox1.Count;
hitung30:='';
teks:='';
```

```
        repeat
        toi:=0;
         begin
           if length(hitung30) < 32 then
             begin
                d:= 31-length(hitung30);
                teks:='';

                if length(listbox1.Items.Strings[b]) <= d then
                  begin
                   hitung30:=hitung30+(listbox1.Items.Strings[b]);
                  end
                  else
                   begin
                    if rightstr(hitung30,1) = ' ' then
                        begin

hitung30:=leftstr(hitung30,(length(hitung30)-1));
                        end;

                     if hitung30<>'' then
                     begin
                      listbox2.Items.Add(hitung30);

                       hitung30:=listbox1.Items.Strings[b];
                     end
                     else
                     hitung30:=listbox1.Items.Strings[b];
                    end;

                  if length(hitung30)>31 then
                     begin
                        buf:=midstr(hitung30,31,(length(hitung30)-31));
                         for e:=1 to length(buf) do
                        begin
                        teks:=teks + midstr(buf,e,1);
                         if length(teks)= 30 then
                         begin
                         listbox4.Items.Add(teks);
                         toi:=toi+1;
                         end;
                        end;
                        if midstr(teks,(30*toi)+1,1) <>' ' then
                        begin


listbox4.Items.Add(midstr(teks,(30*toi)+1,length(teks)));
                           //
listbox4.Items.Add(midstr(hitung30,31,length(hitung30)-30));
                          end;
                          hitung30:= leftstr(hitung30,30);
                        end;
                end;
           end;
          b:=b+1;

          if b=c then
            begin
            if rightstr(hitung30,1) = ' ' then
              begin
               hitung30:=leftstr(hitung30,(length(hitung30)-1));
              end;
```

```
            listbox2.Items.Add(hitung30);
          end;
         until b = c;
        end;
        g:=listbox4.Count;


        if g<>0 then
        begin
        MessageDlg('Kata Anda Lebih dari 30 Karakter (Maka Secara
Otomatis Akan diPotong 30 Karakter)', mtError, [mbOk], 0);
        end;
        end;

//#############################################################################
//PROSEDUR PENGOSONGAN FILE TEKS KETIKA FORM PERTAMA KALI DIBUKA
//#############################################################################
procedure TForm1.FormShow(Sender: TObject);
begin
RichEdit1.Text :='';
end;
//#############################################################################
//PROSEDUR ATUR SERIAL PORT
//#############################################################################
procedure TForm1.Button2Click(Sender: TObject);
begin
comport1.ShowSetupDialog;
end;
//#############################################################################
//PROSEDUR KLIK PADA SCANNING PER 30 KARAKTER
//#############################################################################
procedure TForm1.ListBox2Click(Sender: TObject);
var
        c,edi:string;
        ab       : integer;
        Strab: String;
        a:integer;
begin
        str(listbox2.ItemIndex,c);
        ab:=listbox2.ItemIndex ;
        edit2.Text := listbox2.Items.Strings[ab];
        str(length(edit2.text),edi);
        label6.Caption:=c;
        label4.Caption:=edi;
        edit2.ReadOnly:=True;
        listbox1.ClearSelection;

  Strab := edit2.Text;
  for  a:=1 to length(strab) do
        begin
         strab[a]:=upcase(strab[a]);
        end;
  if length(strab) <> 0 then
  begin
  if length(strab)<31 then
        begin
           repeat
                strab:=strab+' ';
           until length(strab)=31;
        end
        else
        strab:=strab;
```

```
  if comport1.Connected = true then
  begin
 // ComPort1.WriteStr(Str); // string type variable
  ComPort1.Write(Strab[1], Length(Strab)); // no defined type
  end;
  end;
end;

//#############################################################################
//PROSEDUR MENGETAHUI PANJANG KARAKTER YANG AKAN DIKIRIM
//#############################################################################
procedure TForm1.Edit2Change(Sender: TObject);
var
        a:string ;
begin
str(length(edit2.text),a);
label4.Caption := a;
end;

//#############################################################################
//PROSEDUR MERESPON KODE KHUSUS DARI ALAT, MELAKUKAN UPDATE, ATAS TOMBOL
//ATAS DAN TOMBOL BAWAH
//#############################################################################
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
label ending ;
var
  Stra,strh,c: string;
  a,b:integer;
begin
  ComPort1.ReadStr(stra,count);
  timer1.Enabled:=false;

        if listbox2.Focused = true then
        begin

        if stra = 'a' then
        begin

        if listbox2.ItemIndex = (listbox2.Count-1) then
        begin
         listbox2.ItemIndex:=listbox2.ItemIndex;
         end
         else
         listbox2.ItemIndex:=(listbox2.ItemIndex+1);
        end;

        if stra = 'b' then
        begin
        if listbox2.ItemIndex = 0 then
        begin
         listbox2.ItemIndex:=listbox2.ItemIndex;
         end
         else
                listbox2.ItemIndex:=(listbox2.ItemIndex-1);
        end;

        b:=listbox2.ItemIndex;
        edit2.Text:=listbox2.Items.Strings[b];
        Strh := edit2.Text;
        for  a:=1 to length(strh) do
                begin
                strh[a]:=upcase(strh[a]);
                end;
```

```
if length(strh) <> 0 then
        begin
if length(strh)<31 then
        begin
                repeat
                strh:=strh+' ';
                until length(strh)=31;
        end
        else
        strh:=strh;
if comport1.Connected = true then
begin
str(listbox2.ItemIndex,c);
label6.Caption:=c;
timer1.Enabled:=true;
ComPort1.Write(Strh[1], Length(Strh)); // no defined type
end;
end;
end;


if listbox4.Focused = true then
begin

if stra = 'a' then
begin

if listbox4.ItemIndex = (listbox4.Count-1) then
begin
 listbox4.ItemIndex:=listbox4.ItemIndex;
 end
 else
 listbox4.ItemIndex:=(listbox4.ItemIndex+1);
end;

if stra = 'b' then
begin
if listbox4.ItemIndex = 0 then
begin
 listbox4.ItemIndex:=listbox4.ItemIndex;
 end
 else
        listbox4.ItemIndex:=(listbox4.ItemIndex-1);
end;

b:=listbox4.ItemIndex;
edit2.Text:=listbox4.Items.Strings[b];
Strh := edit2.Text;
for  a:=1 to length(strh) do
        begin
        strh[a]:=upcase(strh[a]);
        end;
if length(strh) <> 0 then
        begin
if length(strh)<31 then
        begin
                repeat
                strh:=strh+' ';
                until length(strh)=31;
        end
        else
        strh:=strh;
if comport1.Connected = true then
```

```
        begin
        str(listbox2.ItemIndex,c);
        label6.Caption:=c;
        timer1.Enabled:=true;
        ComPort1.Write(Strh[1], Length(Strh)); // no defined type
        end;
        end;
        end;
  end;
//##############################################################################
//PROSEDUR TOMBOL CONNECT
//##############################################################################
procedure TForm1.Button5Click(Sender: TObject);
begin
comport1.Open;
button5.Enabled:=false;
button6.Enabled:=true;
end;
//##############################################################################
//PROSEDUR TOMBOL DISCONNECT
//##############################################################################
procedure TForm1.Button6Click(Sender: TObject);
begin
comport1.Close;
button5.Enabled := true;
button6.Enabled := false;
end;
//##############################################################################
procedure TForm1.FormCreate(Sender: TObject);
begin
button6.Enabled:=false;
end;
//##############################################################################
//PROSEDUR TIMER PENGIRIMAN
//##############################################################################
procedure TForm1.Timer1Timer(Sender: TObject);
begin
timer1.Interval:=10000;
end;
//##############################################################################
//PROSEDUR PENGIRIMAN KATA YANG TERPOTONG
//##############################################################################
procedure TForm1.ListBox1Click(Sender: TObject);
begin
    listbox1.ClearSelection;
    listbox2.SetFocus;
end;

procedure TForm1.ListBox4Click(Sender: TObject);
var
        c,edi:string;
        ab      : integer;
        Strab: String;
        a:integer;
begin
        str(listbox4.ItemIndex,c);
        ab:=listbox4.ItemIndex ;
        edit2.Text := listbox4.Items.Strings[ab];
        str(length(edit2.text),edi);
        label6.Caption:=c;
        label4.Caption:=edi;
        edit2.ReadOnly:=True;
        listbox1.ClearSelection;
```

```
  Strab := edit2.Text;
  for  a:=1 to length(strab) do
       begin
        strab[a]:=upcase(strab[a]);
       end;
  if length(strab) <> 0 then
  begin
  if length(strab)<31 then
       begin
          repeat
               strab:=strab+' ';
          until length(strab)=31;
       end
       else
       strab:=strab;
  if comport1.Connected = true then
  begin
 // ComPort1.WriteStr(Str); // string type variable
  ComPort1.Write(Strab[1], Length(Strab)); // no defined type
  end;
  end;
end;
end.
```

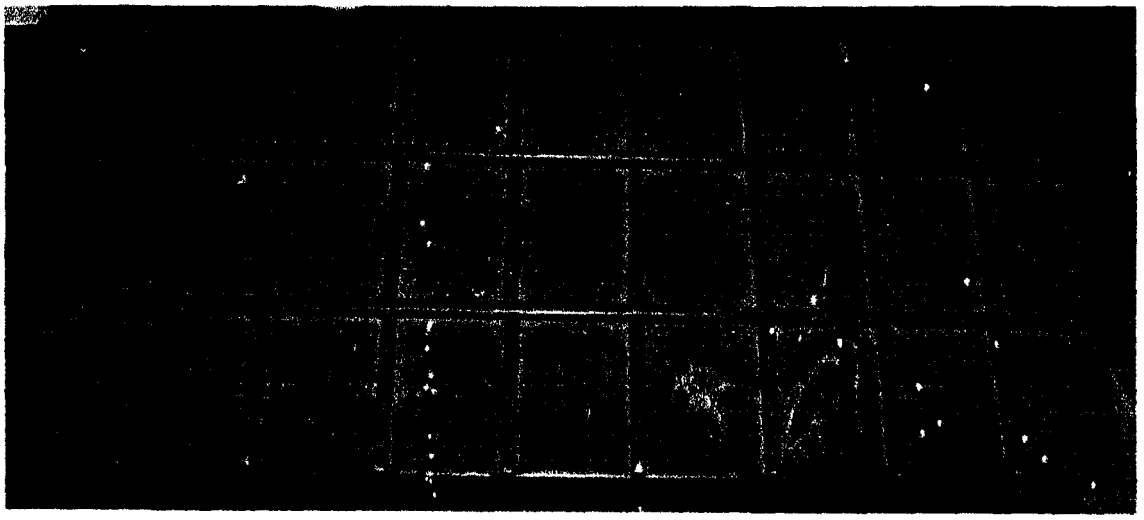
LAMPIRAN D

HASIL KUNJUNGAN DI YAYASAN PENDIDIKAN ANAK BUTA

ALAT PEMBELAJARAN PENGENALAN HURUF BRAILLE DI YAYASAN PENDIDIKAN ANAK BUTA JL.TEGALSARI NO 56

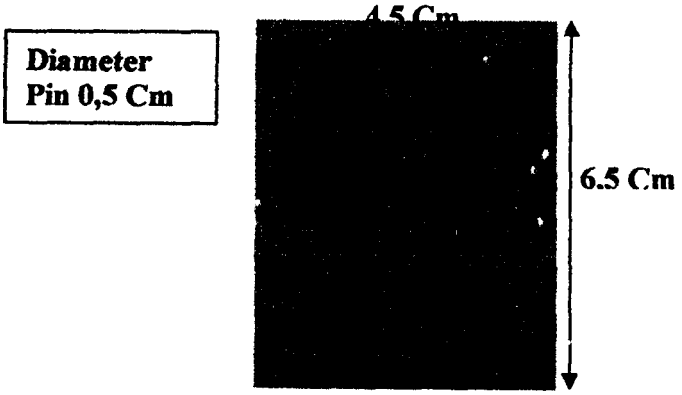
Alat pembelajaran untuk mengenalkan huruf Braille bagi penyandang tuna netra yang ada di YPAB (Yayasan Pendidikan Anak Buta) Jl. Tegalsari no. 56 terbagi atas dua tingkatan yaitu :

1. Tingkat Pemula



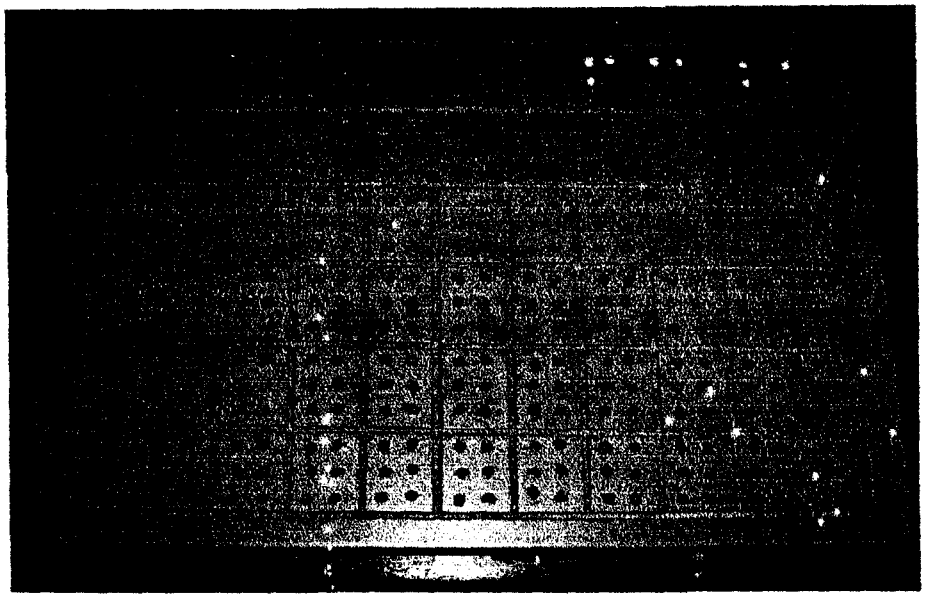
Gambar L.16. Alat Pengenalan Braille Tingkat Pemula

Alat ini terbuat dari kayu dan terdiri dari 27 sel kode Braille. Ukuran setiap sel dari alat diatas selengkapnya dapat dilihat pada gambar L.17.



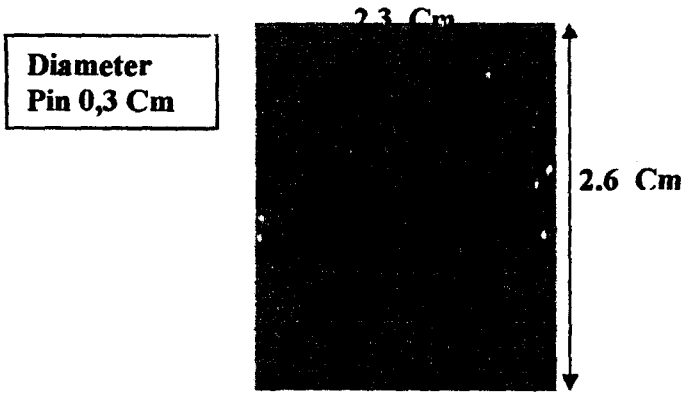
Gambar L.17. Ukuran Alat Pengenalan Braille Tingkat Pemula

2. Tingkat Lanjutan



Gambar L.18. Alat Pengenalan Braille Tingkat Lanjutan

Alat ini juga terbuat dari kayu dan memiliki 66 sel kode Braille. Ukuran setiap sel dari alat diatas selengkapnya dapat dilihat pada gambar L.19.



Gambar L.19. Ukuran Alat Pengenalan Braille Tingkat Lanjutan

Universitas Katolik Widya Mandala Surabaya
Fakultas Teknik
JURUSAN TEKNIK ELEKTRO
Jalan Kalijudan 37 Surabaya 60114 Indonesia – Telp. (031) 3891264 psw.103 Faks. 3891267
Website : http://www.eng.wima.ac.id Email: elektro@mail.wima.ac.id

Nomor : 0129 /WM05.1/T/2006 28 September 2006
Lampiran : -
Hal : **PERMOHONAN IJIN OBSERVASI**

Kepada : Yth. **Kepala**
Sekolah Pendidikan Tuna Netra
Jl. Tegalsari 56
Surabaya

Dengan hormat,

Sehubungan dengan penyusunan tugas akhir yang harus dilaksanakan oleh setiap mahasiswa Jurusan Teknik Elektro Fakultas Teknik Universitas Katolik Widya Mandala Surabaya, maka dengan hormat kami mohonkan ijin bagi mahasiswa kami berikut ini untuk melakukan observasi (bagaimana cara membantu anak tuna netra agar dapat membaca teks yang ada pada komputer/PC) yaitu :

FRANSISCUS XAVERIUS EVEN LIMBA / 5103002070

Adapun **hasil observasi** tersebut nantinya akan digunakan untuk mendukung penyusunan tugas akhir yang bersangkutan dengan judul :

* *Pembangkit kode braille dinamis dengan sumber teks dari PC.*

Demikian permohonan ini kami sampaikan, atas perhatian dan ijin yang akan diberikan, sebelumnya kami ucapkan terima kasih.

Hormat kami,
a.n. Ketua Jurusan,
Sekretaris Jurusan,

Andrew Joewono,MT.
NIK. 511.97.0291

Questioner

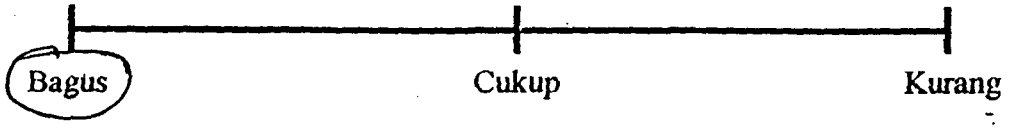
"Pembangkit Kode Braille Dinamis Dengan Sumber Teks dari PC"
Di
Yayasan Pendidikan Anak Buta
Jl. Tegalsari 56, Surabaya

1. Menurut anda apakah ide pembuatan alat ini menarik ?
   - ☑ Sangat Menarik
   - ☐ Menarik
   - ☐ Biasa saja
   - ☐ Tidak Menarik
2. Menurut anda apakah pembuatan alat ini membantu tuna netra ?
   - ☑ Sangat Membantu
   - ☐ Membantu
   - ☐ Cukup Membantu
   - ☐ Tidak Membantu
3. Bagaimana jika sistem ini diterapkan dan dikembangkan ?
   - ☑ Setuju
   - ☐ Tidak, alasan :

     .....

     .....

     .....
4. Menurut anda bagaimana penilaian anda tentang "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC" ini ?



5. Kritik dan Saran

Saran bentuk hurufnya sesuaikan dengan standart huruf braille yang Asli

~ TERIMA KASIH ~

TUHAN YESUS MEMBERKATI

Mengetahui Kepala Sekolah

SLB

SLB/A YPAB SURABAYA

SURABAYA

Drs Mujiraharjo

30 - 9 - 06

Yata Pribadi SE

Questioner

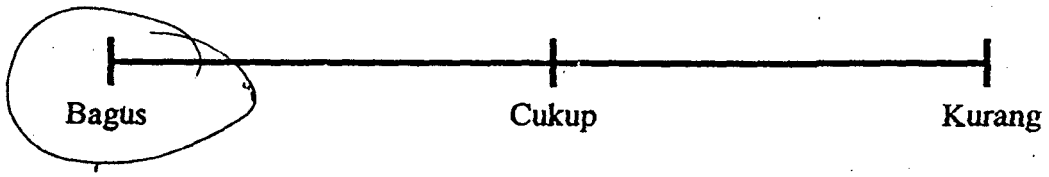
"Pembangkit Kode Braille Dinamis Dengan Sumber Teks dari PC"
Di
Yayasan Pendidikan Anak Buta
Jl. Tegalsari 56, Surabaya

1. Menurut anda apakah ide pembuatan alat ini menarik ?
   - ☒ Sangat Menarik
   - ☐ Menarik
   - ☐ Biasa saja
   - ☐ Tidak Menarik
2. Menurut anda apakah pembuatan alat ini membantu tuna netra ?
   - ☐ Sangat Membantu
   - ☒ Membantu
   - ☐ Cukup Membantu
   - ☐ Tidak Membantu
3. Bagaimana jika sistem ini diterapkan dan dikembangkan ?
   - ☒ Setuju
   - ☐ Tidak, alasan :

     .....

     .....

     .....
4. Menurut anda bagaimana penilaian anda tentang "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC" ini ?



5. **Kritik dan Saran**

Diperjelas lagi timbul atau nggaknya. Lebih disempurnakan lagi

~ TERIMA KASIH ~
TUHAN YESUS MEMBERKATI

Mengetahui
Kepala Sekolah

SLB
SLB / A YPAB SURABAYA
SURABAYA

Drs. Muji Raharjo

30-9-06

Tantri Meliawan, SPd.

Questioner

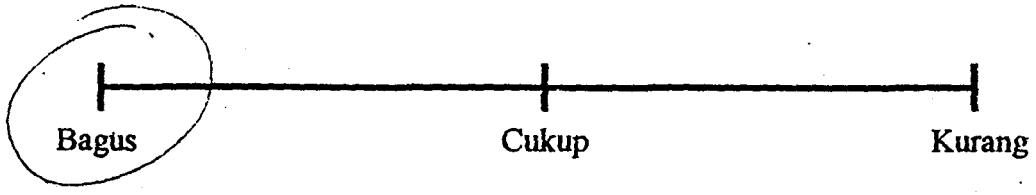
"Pembangkit Kode Braille Dinamis Dengan Sumber Teks dari PC"
Di
Yayasan Pendidikan Anak Buta
Jl. Tegalsari 56, Surabaya

1. Menurut anda apakah ide pembuatan alat ini menarik ?
   - ☑ Sangat Menarik
   - ☐ Menarik
   - ☐ Biasa saja
   - ☐ Tidak Menarik
2. Menurut anda apakah pembuatan alat ini membantu tuna netra ?
   - ☑ Sangat Membantu
   - ☐ Membantu
   - ☐ Cukup Membantu
   - ☐ Tidak Membantu
3. Bagaimana jika sistem ini diterapkan dan dikembangkan ?
   - ☑ Setuju
   - ☐ Tidak, alasan .

     .....

     .....

     .....
4. Menurut anda bagaimana penilaian anda tentang "Pembangkit Kode Braille Dinamis dengan Sumber Teks dari PC" ini ?



5. **Kritik dan Saran**

1. Bentuk tampilan huruf sebaiknya dibuat lebih kecil

2. Bila memungkinkan dibuat dalam jumlah huruf yg lebih banyak.

3. Upayakan permukaan titik yang timbul dan tidak timbul ada perbedaan yang jelas

4. Untuk tunanetra pemula alat ini sangat bagus

~ TERIMA KASIH ~
TUHAN YESUS MEMBERKATI

Kepala SLB-A YPAB

SLB
SURAKARTA

Drs Mugi Waluyo

FAIRCHILD
SEMICONDUCTOR™

April 1986
Revised March 2000

DM74LS373 • DM74LS374
3-STATE Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

General Description

These 8-bit registers feature totem-pole 3-STATE outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance state and increased high-logic level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the DM74LS373 are transparent D-type latches meaning that while the enable (G) is HIGH the Q outputs will follow the data (D) inputs. When the enable is taken LOW the output will be latched at the level of the data that was set up.

The eight flip-flops of the DM74LS374 are edge-triggered D-type flip flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were set up at the D inputs.

A buffered output control input can be used to place the eight outputs in either a normal logic state (HIGH or LOW logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are OFF.

Features

- Choice of 8 latches or 8 D-type flip-flops in a single package
- 3-STATE bus-driving outputs
- Full parallel-access for loading
- Buffered control inputs
- P-N-P inputs reduce D-C loading on data lines

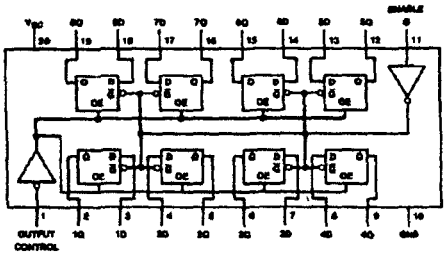
Ordering Code:

Order Number	Package Number	Package Description
DM74LS373WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300 Wide
DM74LS373SJ	M20D	20-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS373N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
DM74LS374WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300 Wide
DM74LS374SJ	M20D	20-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
IDM29901NC	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

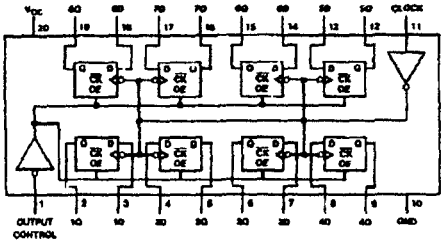
Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagrams

DM74LS373



DM74LS374



Function Tables

DM74LS373

Output Control	Enable G	D	Output
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

DM74LS374

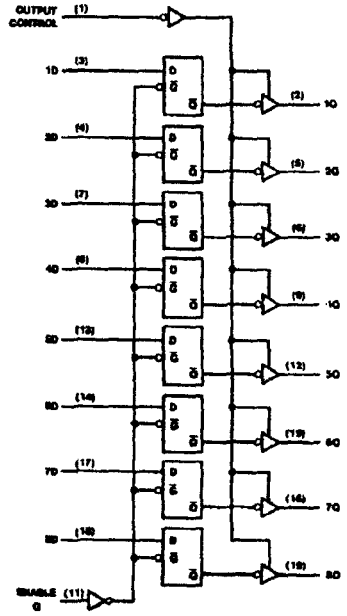
Output Control	Clock	D	Output
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

H = HIGH Level (Steady State) L = LOW Level (Steady State) X = Don't Care Z = High Impedance State

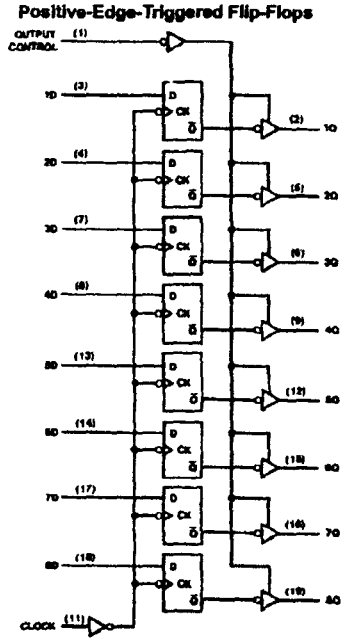
↑ = Transition from LOW-to-HIGH level Q_0 = The level of the output before steady-state input conditions were established.

Logic Diagrams

DM74LS373
Transparent Latches



DM74LS374
Positive-Edge-Triggered Flip-Flops



Absolute Maximum Ratings(Note 1)

Supply Voltage	7V
Input Voltage	7V
Storage Temperature Range	−65°C to +150°C
Operating Free Air Temperature Range	0°C to +70°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

DM74LS373 Recommended Operating Conditions

Symbol	Parameter		Min	Nom	Max	Units
V_{CC}	Supply Voltage		4.75	5	5.25	V
V_{IH}	HIGH Level Input Voltage		2			V
V_{IL}	LOW Level Input Voltage				0.8	V
I_{OH}	HIGH Level Output Current				−2.6	mA
I_{OL}	LOW Level Output Current				24	mA
t_W	Pulse Width	Enable HIGH	15			ns
	(Note 3)	Enable LOW	15			
t_{SU}	Data Setup Time (Note 2) (Note 3)		5↓			ns
t_H	Data Hold Time (Note 2) (Note 3)		20↓			ns
T_A	Free Air Operating Temperature		0		70	°C

Note 2: The symbol (↓) indicates the falling edge of the clock pulse is used for reference.

Note 3: $T_A = 25°C$ and $V_{CC} = 5V$.

DM74LS373 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 4)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	HIGH Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	2.4	3.1		V
V_{OL}	LOW Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 12 mA, V_{CC} = Min			0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	HIGH Level Input Current	V_{CC} = Max, V_I = 2.7V			20	µA
I_{IL}	LOW Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{OZH}	Off-State Output Current with HIGH Level Output Voltage Applied	V_{CC} = Max, V_O = 2.7V, V_{IH} = Min, V_{IL} = Max			20	µA
I_{OZL}	Off-State Output Current with LOW Level Output Voltage Applied	V_{CC} = Max, V_O = 0.4V, V_{IH} = Min, V_{IL} = Max			−20	µA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 5)	−50		−225	mA
I_{CC}	Supply Current	V_{CC} = Max, OC = 4.5V, D_n, Enable = GND		24	40	mA

Note 4: All typicals are at $V_{CC} = 5V$, $T_A = 25°C$.

Note 5: Not more than one output should be shorted at a time, and the duration should not exceed one second.

DM74LS373 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	From (Input) To (Output)	R_L = 667Ω				Units
			C_L = 45 pF		C_L = 150 pF		
			Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Data to Q		18		26	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Data to Q		18		27	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Enable to Q		30		38	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Enable to Q		30		36	ns
t_{PZH}	Output Enable Time to HIGH Level Output	Output Control to Any Q		28		36	ns
t_{PZL}	Output Enable Time to LOW Level Output	Output Control to Any Q		36		50	ns
t_{PHZ}	Output Disable Time from HIGH Level Output (Note 6)	Output Control to Any Q		20			ns
t_{PLZ}	Output Disable Time from LOW Level Output (Note 6)	Output Control to Any Q		25			ns

Note 6: C_L = 5 pF.

DM74LS374 Recommended Operating Conditions

Symbol	Parameter		Min	Nom	Max	Units
V_{CC}	Supply Voltage		4.75	5	5.25	V
V_{IH}	HIGH Level Input Voltage		2			V
V_{IL}	LOW Level Input Voltage				0.8	V
I_{OH}	HIGH Level Output Current				-2.6	mA
I_{OL}	LOW Level Output Current				24	mA
t_W	Pulse Width (Note 8)	Clock HIGH	15			ns
		Clock LOW	15			
t_{SU}	Data Setup Time (Note 7) (Note 8)		20↑			ns
t_H	Data Hold Time (Note 7) (Note 8)		1↑			ns
T_A	Free Air Operating Temperature		0		70	°C

Note 7: The symbol (↑) indicates the rising edge of the clock pulse is used for reference.

Note 8: T_A = 25°C and V_{CC} = 5V.

DM74LS374 Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 9)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	HIGH Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max, V_{IH} = Min	2.4	3.1		V
V_{OL}	LOW Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 12 mA, V_{CC} = Min		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	HIGH Level Input Current	V_{CC} = Max, V_I = 2.7V			20	µA
I_{IL}	LOW Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{OZH}	Off-State Output Current with HIGH Level Output Voltage Applied	V_{CC} = Max, V_O = 2.7V, V_{IH} = Min, V_{IL} = Max			20	µA
I_{OZL}	Off-State Output Current with LOW Level Output Voltage Applied	V_{CC} = Max, V_O = 0.4V, V_{IH} = Min, V_{IL} = Max			−20	µA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 10)	−50		−225	mA
I_{CC}	Supply Current	V_{CC} = Max, D_n = GND, OC = 4.5V		27	45	mA

Note 9: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 10: Not more than one output should be shorted at a time, and the duration should not exceed one second.

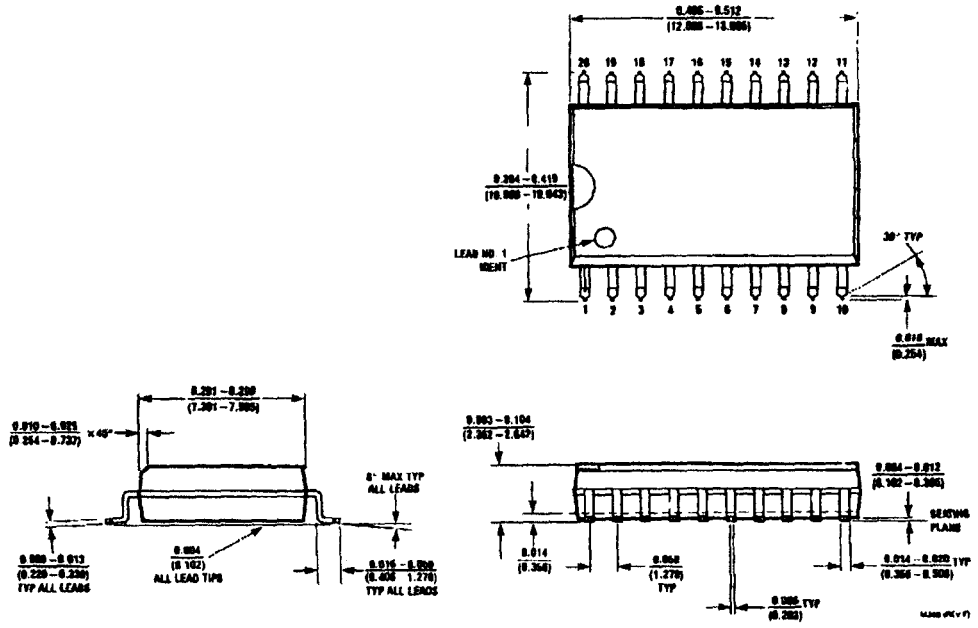
DM74LS374 Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	R_L = 667Ω				Units
		C_L = 45 pF		C_L = 150 pF		
		Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency	35		20		MHz
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output		28		32	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output		28		38	ns
t_{PZH}	Output Enable Time to HIGH Level Output		28		44	ns
t_{PZL}	Output Enable Time to LOW Level Output		28		44	ns
t_{PHZ}	Output Disable Time from HIGH Level Output (Note 11)		20			ns
t_{PLZ}	Output Disable Time from LOW Level Output (Note 11)		25			ns

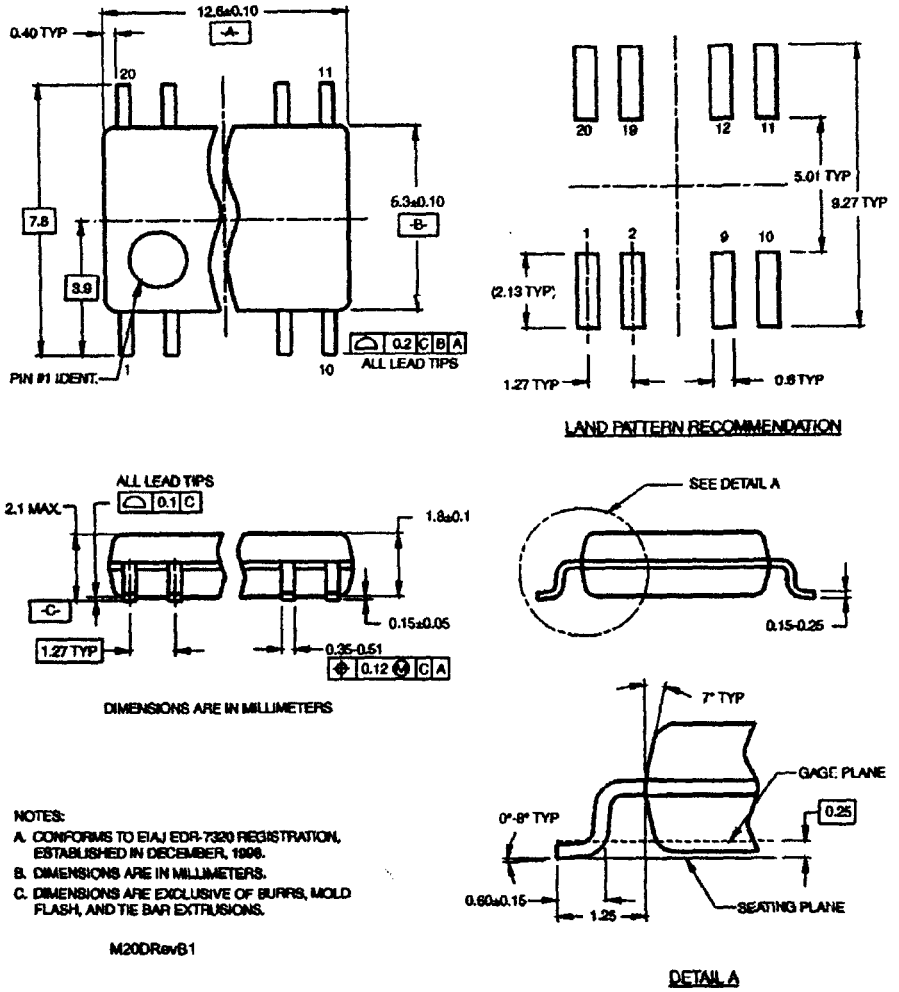
Note 11: C_L = 5 pF.

Physical Dimensions inches (millimeters) unless otherwise noted



20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300 Wide
Package Number M20B

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



NOTES:

A. CONFORMS TO EIAJ EDR-7320 REGISTRATION, ESTABLISHED IN DECEMBER, 1996.

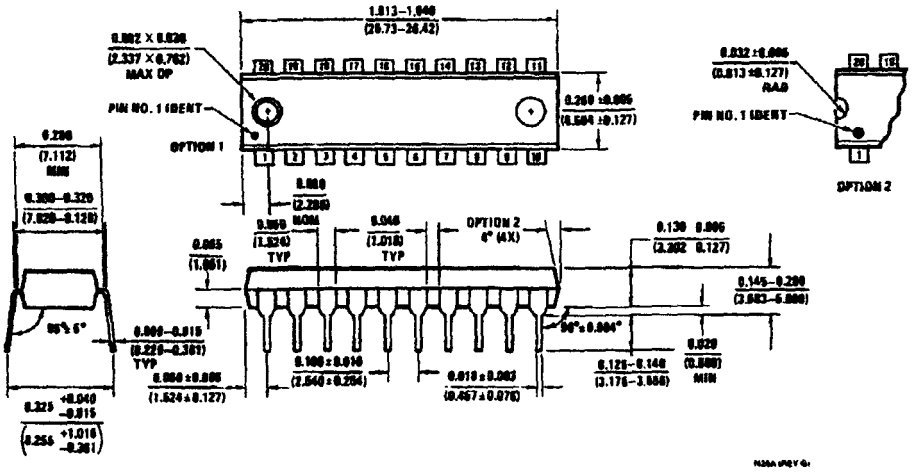
B. DIMENSIONS ARE IN MILLIMETERS.

C. DIMENSIONS ARE EXCLUSIVE OF BURRS, MOLD FLASH, AND TIE BAR EXTRUSIONS.

M20DRevB1

20-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
Package Number M20D

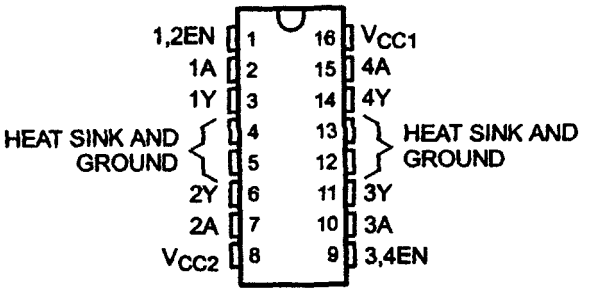
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



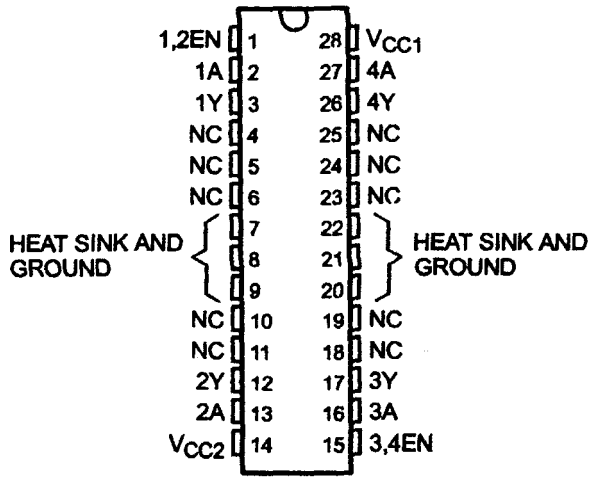
20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
Package Number N20A

- **Featuring Unitrode L293 and L293D Products Now From Texas Instruments**
- **Wide Supply-Voltage Range: 4.5 V to 36 V**
- **Separate Input-Logic Supply**
- **Internal ESD Protection**
- **Thermal Shutdown**
- **High-Noise-Immunity Inputs**
- **Functional Replacements for SGS L293 and SGS L293D**
- **Output Current 1 A Per Channel (600 mA for L293D)**
- **Peak Output Current 2 A Per Channel (1.2 A for L293D)**
- **Output Clamp Diodes for Inductive Transient Suppression (L293D)**

N, NE PACKAGE
(TOP VIEW)



DWP PACKAGE
(TOP VIEW)



description

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

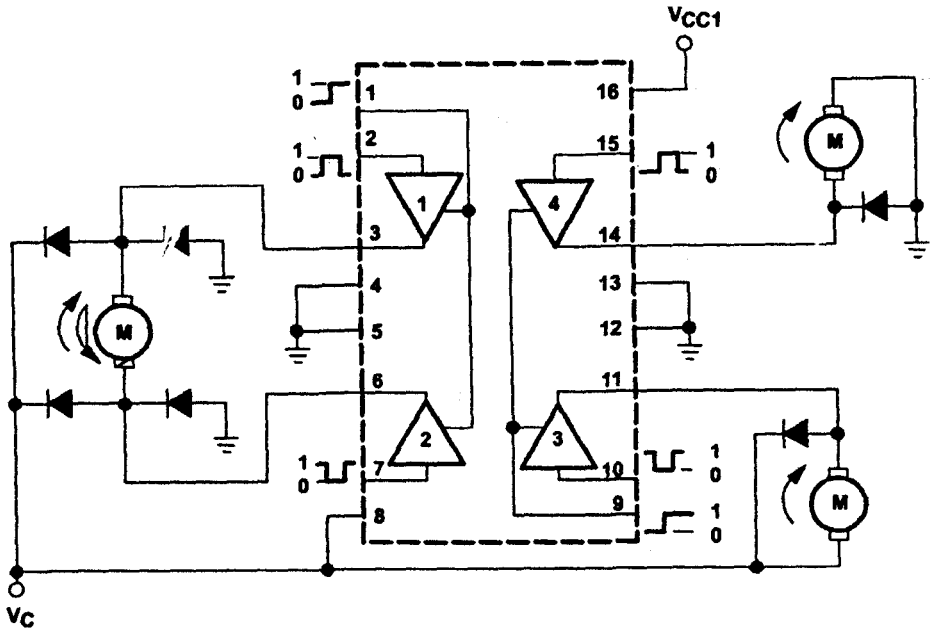
All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

On the L293, external high-speed output clamp diodes should be used for inductive transient suppression.

A V_{CC1} terminal, separate from V_{CC2}, is provided for the logic inputs to minimize device power dissipation.

The L293and L293D are characterized for operation from 0°C to 70°C.

block diagram



NOTE: Output diodes are internal in L293D.

TEXAS INSTRUMENTS AVAILABLE OPTIONS

T_A	PACKAGE
	PLASTIC DIP (NE)
0°C to 70°C	L293NE L293DNE

Unitrode Products from Texas Instruments

AVAILABLE OPTIONS

T_A	PACKAGED DEVICES	
	SMALL OUTLINE (DWP)	PLASTIC DIP (N)
0°C to 70°C	L293DWP L293DDWP	L293N L293DN

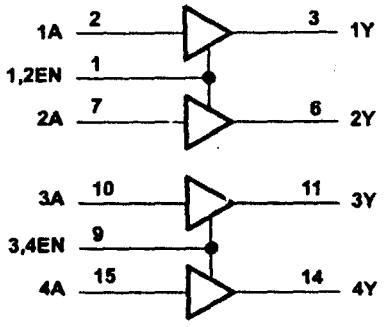
The DWP package is available taped and reeled. Add the suffix TR to device type (e.g., L293DWPTR).

FUNCTION TABLE
(each driver)

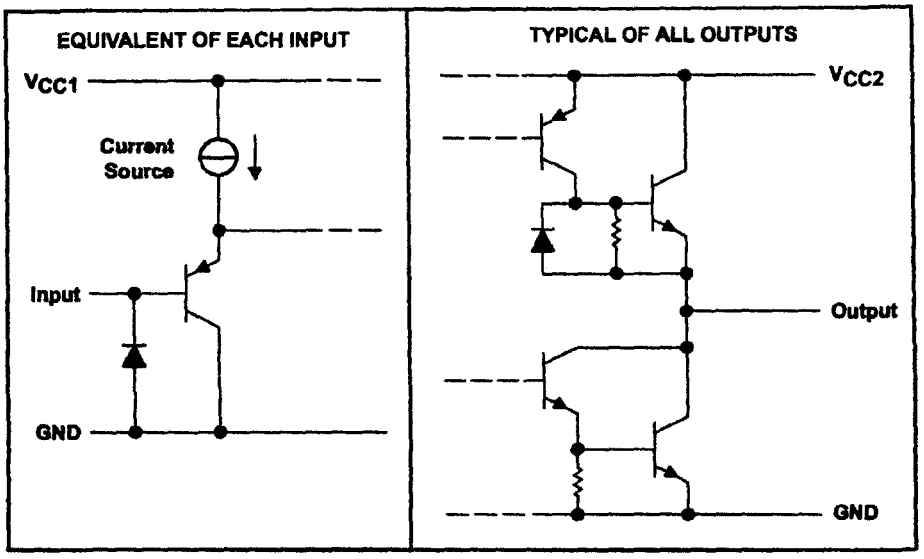
INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant,
Z = high impedance (off)
† In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

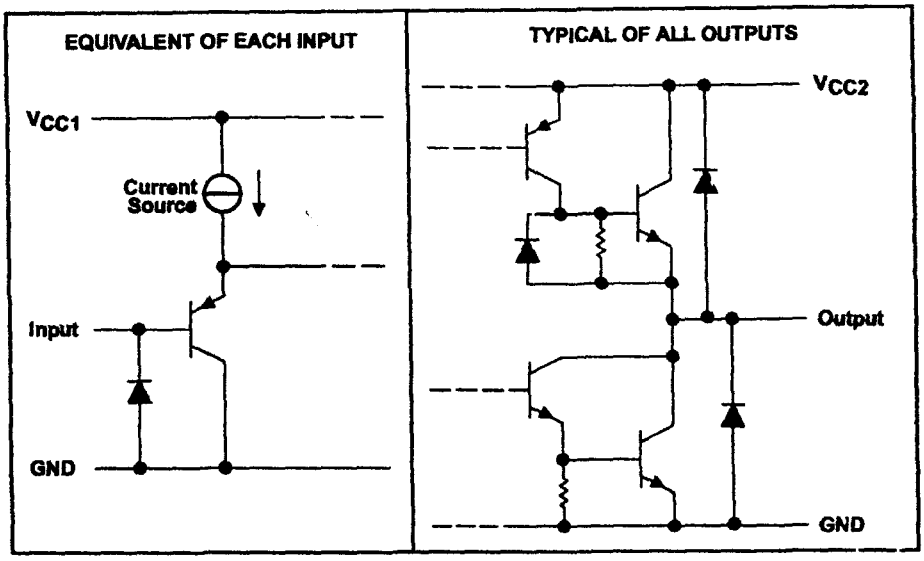
logic diagram



schematics of inputs and outputs (L293)



schematics of inputs and outputs (L293D)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC1} (see Note 1)	36 V
Output supply voltage, V_{CC2}	36 V
Input voltage, V_I	7 V
Output voltage range, V_O	–3 V to V_{CC2} + 3 V
Peak output current, I_O (nonrepetitive, t ≤ 5 ms): L293	±2 A
Peak output current, I_O (nonrepetitive, t ≤ 100 µs): L293D	±1.2 A
Continuous output current, I_O: L293	±1 A
Continuous output current, I_O: L293D	±600 mA
Continuous total dissipation at (or below) 25°C free-air temperature (see Notes 2 and 3)	2075 mW
Continuous total dissipation at 80°C case temperature (see Note 3)	5000 mW
Maximum junction temperature, T_J	150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, T_{stg}	–65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values are with respect to the network ground terminal.
2. For operation above 25°C free-air temperature, derate linearly at the rate of 16.6 mW/°C.
3. For operation above 25°C case temperature, derate linearly at the rate of 71.4 mW/°C. Due to variations in individual device electrical characteristics and thermal resistance, the built-in thermal overload protection may be activated at power levels slightly above or below the rated dissipation.

recommended operating conditions

			MIN	MAX	UNIT
	Supply voltage	V_{CC1}	4.5	7	V
		V_{CC2}	V_{CC1}	36	
V_{IH}	High-level input voltage	$V_{CC1} \le 7$ V	2.3	V_{CC1}	V
		$V_{CC1} \ge 7$ V	2.3	7	V
V_{IL}	Low-level output voltage		–0.3†	1.5	V
T_A	Operating free-air temperature		0	70	°C

† The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

electrical characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER			TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage		L293: I_{OH} = –1 A L293D: I_{OH} = –0.6 A		V_{CC2}–1.8	V_{CC2}–1.4		V
V_{OL}	Low-level output voltage		L293: I_{OL} = 1 A L293D: I_{OL} = 0.6 A			1.2	1.8	V
V_{OKH}	High-level output clamp voltage		L293D: I_{OK} = –0.6 A			V_{CC2} + 1.3		V
V_{OKL}	Low-level output clamp voltage		L293D: I_{OK} = 0.6 A			1.3		V
I_{IH}	High-level input current	A	V_I = 7 V			0.2	100	µA
		EN				0.2	10	
I_{IL}	Low-level input current	A	V_I = 0			–3	–10	µA
		EN				–2	–100	
I_{CC1}	Logic supply current		I_O = 0	All outputs at high level		13	22	mA
				All outputs at low level		35	60	
				All outputs at high impedance		8	24	
I_{CC2}	Output supply current		I_O = 0	All outputs at high level		14	24	mA
				All outputs at low level		2	6	
				All outputs at high impedance		2	4	

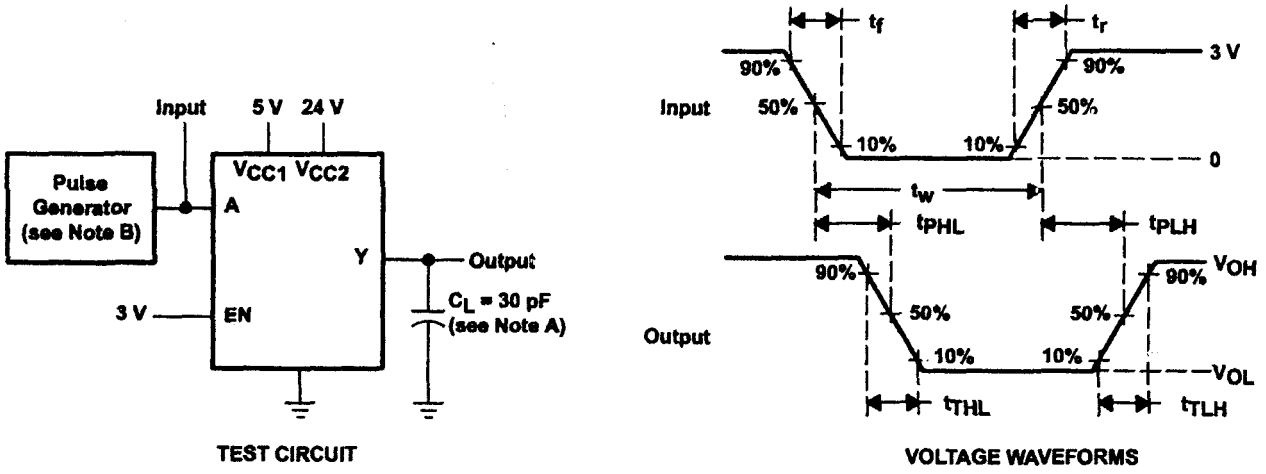
switching characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	L293NE, L293DNE			UNIT
			MIN	TYP	MAX	
t_{PLH}	Propagation delay time, low-to-high-level output from A input	C_L = 30 pF, See Figure 1		800		ns
t_{PHL}	Propagation delay time, high-to-low-level output from A input			400		ns
t_{TLH}	Transition time, low-to-high-level output			300		ns
t_{THL}	Transition time, high-to-low-level output			300		ns

switching characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER		TEST CONDITIONS	L293DWP, L293N L293DDWP, L293DN			UNIT
			MIN	TYP	MAX	
t_{PLH}	Propagation delay time, low-to-high-level output from A input	C_L = 30 pF, See Figure 1		750		ns
t_{PHL}	Propagation delay time, high-to-low-level output from A input			200		ns
t_{TLH}	Transition time, low-to-high-level output			100		ns
t_{THL}	Transition time, high-to-low-level output			350		ns

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT

VOLTAGE WAVEFORMS

NOTES: A. C_L includes probe and jig capacitance.
B. The pulse generator has the following characteristics: $t_r \leq 10$ ns, $t_f \leq 10$ ns, $t_w = 10$ µs, PRR = 5 kHz, $Z_O = 50\ \Omega$.

Figure 1. Test Circuit and Voltage Waveforms

APPLICATION INFORMATION

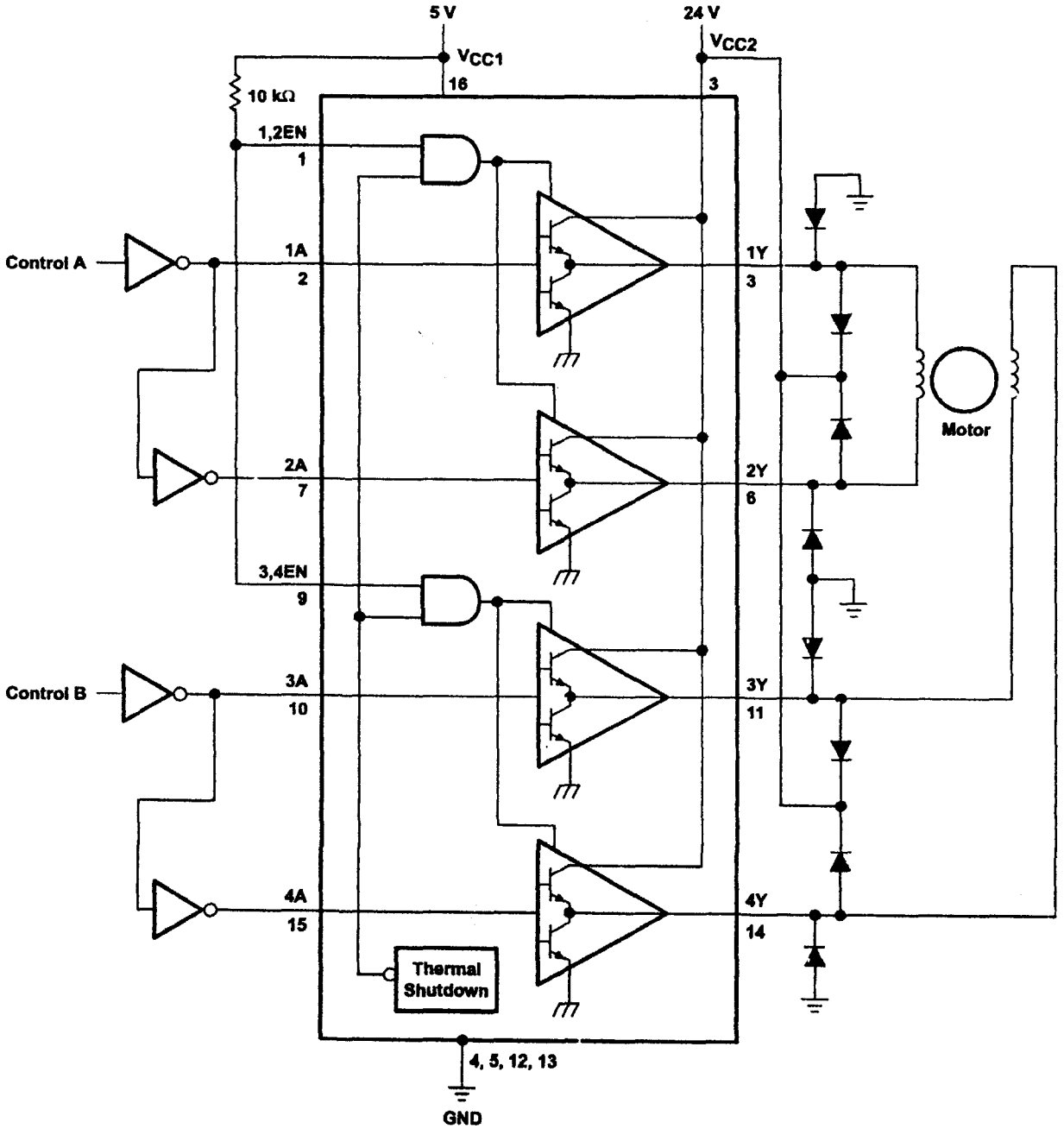


Figure 2. Two-Phase Motor Driver (L293)

APPLICATION INFORMATION

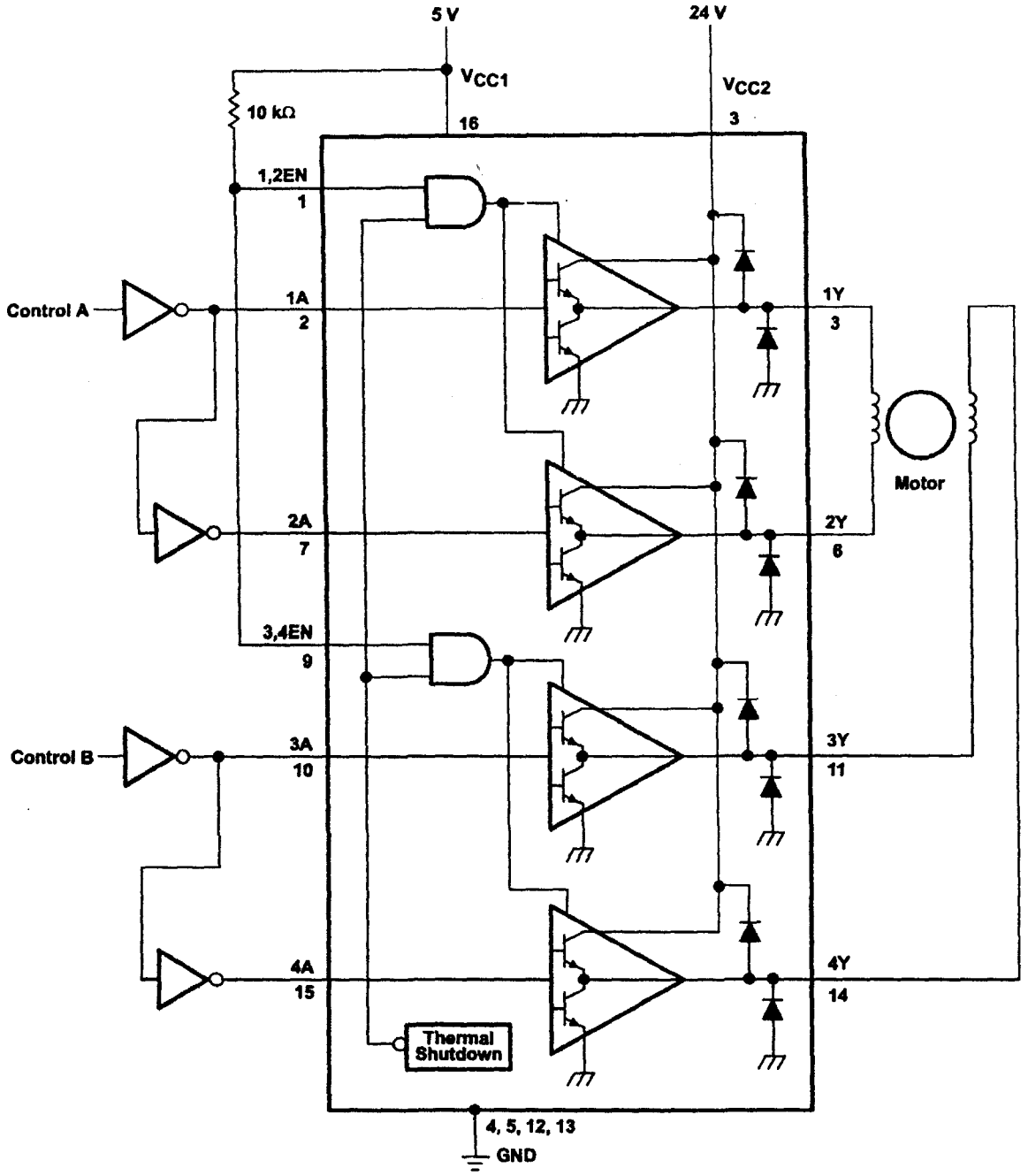
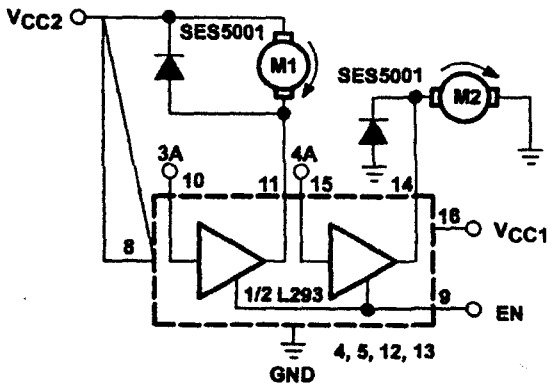


Figure 3. Two-Phase Motor Driver (L293D)

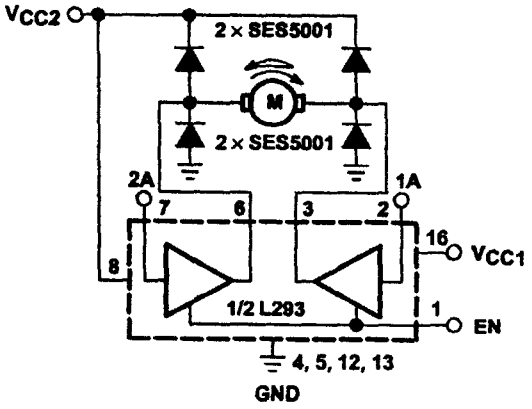
APPLICATION INFORMATION



EN	3A	M1	4A	M2
H	H	Fast motor stop	H	Run
H	L	Run	L	Fast motor stop
L	X	Free-running motor stop	X	Free-running motor stop

L = low, H = high, X = don't care

Figure 4. DC Motor Controls (connections to ground and to supply voltage)

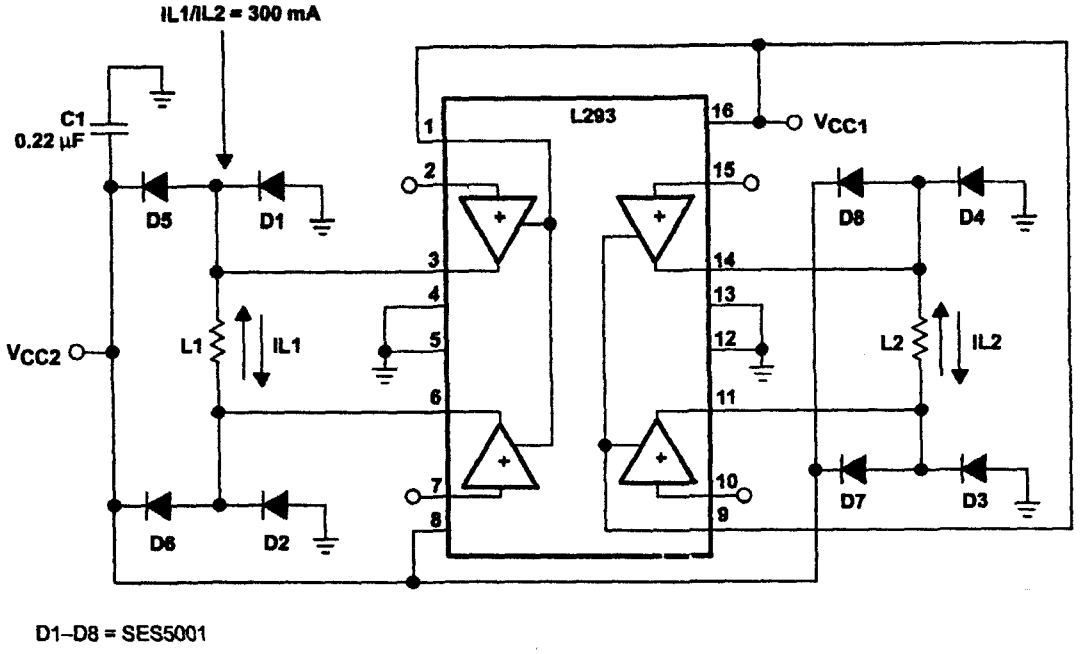


EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

Figure 5. Bidirectional DC Motor Control

APPLICATION INFORMATION



D1–D8 = SES5001

Figure 6. Bipolar Stepping-Motor Control

mounting instructions

The Rthj-amp of the L293 can be reduced by soldering the GND pins to a suitable copper area of the printed circuit board or to an external heatsink.

Figure 9 shows the maximum package power P_{TOT} and the θ_{JA} as a function of the side ℓ of two equal square copper areas having a thickness of 35 µm (see Figure 7). In addition, an external heat sink can be used (see Figure 8).

During soldering, the pin temperature must not exceed 260°C, and the soldering time must not be longer than 12 seconds.

The external heatsink or printed circuit copper area must be connected to electrical ground.

APPLICATION INFORMATION

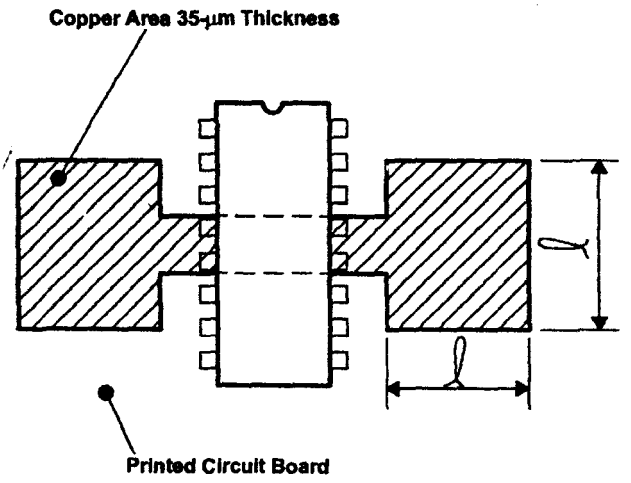


Figure 7. Example of Printed Circuit Board Copper Area (used as heat sink)

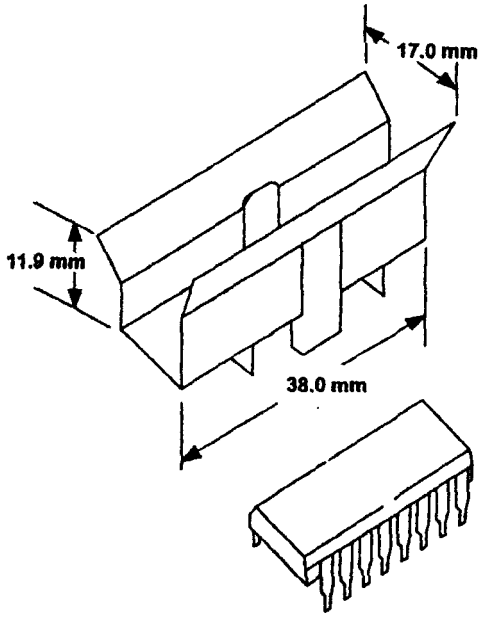


Figure 8. External Heat Sink Mounting Example (θ_{JA} = 25°C/W)

APPLICATION INFORMATION

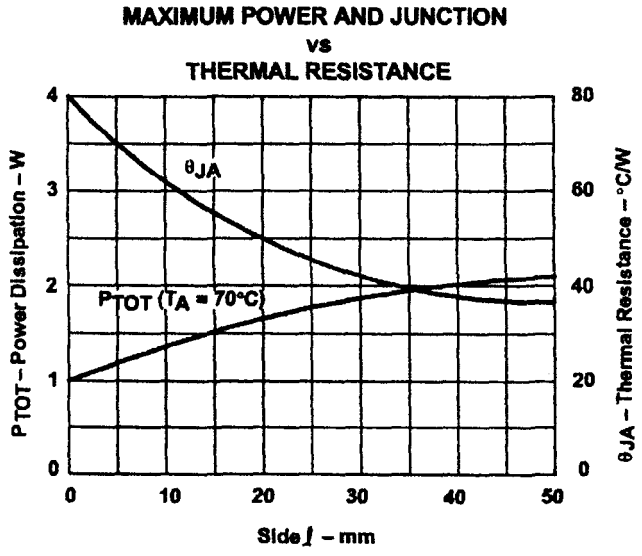


Figure 9

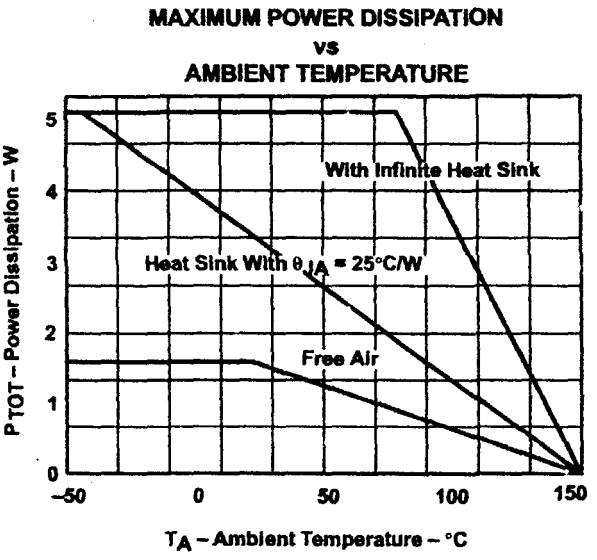


Figure 10

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

BIODATA PENULIS

Nama : Fransiscus Xaverius Even Limba

NRP : 5103002070

Tempat / Tgl. Lahir : Makassar, 10 April 1984

Agama : Kristen

Alamat : Sutorejo Prima Selatan III PG 8/7

Surabaya 60113

Riwayat Pendidikan :

- Tahun 1996, lulus SDK Kalam Kudus, Makassar.
- Tahun 1999, lulus SLTP Kristen Gamaliel, Makassar.
- Tahun 2002, lulus SMU YPPI – I, Surabaya.
- Tahun 2002 hingga biodata ini ditulis tercatat sebagai mahasiswa Fakultas Teknik Jurusan Teknik Elektro di Universitas Katolik Widya Mandala Surabaya.