Algoritma Program

1. Penekanan tombol 'Setting'
2. Penekanan tombol 'Connect'
3. Penekanan tombol 'Check' untuk mengetahui

   Slave mana yang aktif
4. Apabila tombol 'Capture1 dan/atau capture 2 ' aktif berarti

   slave 1dan/atau slave 2 dalam keadaan aktif
5. Untuk melihat tampilan Slave 1 atau 2 maka dapat ditekan

   tombol 'Capture1 atau capture 2 '

Algoritma Setting

1. Inisialisasi terhadap serial COM1 atau COM2 yang dalam

   keadaan aktif dan baudrate telah di fix kan pada

   115200 (dengan bantuan library CPdrv)

Algoritma Connect

1. Melakukan set untuk kondisi awal dari masing-masing button.

   apakah aktif ataukah tidak

Algoritma Check

1. Apabila tombol Cek ditekan maka akan dikirimkan karakter

   ke slave 1 (#~@M1?)

2. Delay selama 2 dtk agar tidak terjadi data collision
3. Selama delay slave akan melakukan pengecekan, apabila dia slave 1 maka akan dilakukan pengiriman karakter (#~@S01) ke master sebagai tanda bahwa slave 1 dalam keadaan aktif
4. Apabila delay selesai maka dikirim karakter ke slave 2
5. Apabila slave 2 dalam keadaan aktif maka akan dikirim karakter (#~@S02) sebagai tanda bahwa slave 2 aktif
6. Apabila master menerima karakter-karakter tersebut maka tombol capture akan dikondisikan aktif

Algoritma Capture

1. Apabila tombol capture 1/2 ditekan maka master akan mengirim karakter (#~@M1>) untuk meminta besar file ke slave bersangkutan
2. Apabila slave menerima karakter tersebut, slave bersangkutan akan melakukan proses capture desktop dari windows (*GetwindowDC*) dan mengirimkan ukuran besar file ke master (dengan karakter #~@B0)
3. Setelah master menerima besar file maka dikirim perintah agar slave bersangkutan mengirim file hasil capture (dengan karakter #~@M1=)

4. Slave akan menyiapkan array 8192, dimana 1-5 diisi header (#~@D0), 6-8191 diisi data (ukuran data 8186), array ke 8192 diisi penjumlahan total dari data tersebut (sebagai checksum).
5. Slave akan mengirim paket tersebut ke master
6. Master akan mengambil data dari buffer selama itu juga dilakukan penjumlahan terhadap nilai data tersebut
7. Bila data telah terambil semua dari buffer , maka akan dicocokkan penjumlahan data total oleh master dan penjumlahan data total oleh slave.

   Bila sama maka dimulai menulis data tersebut dalam file

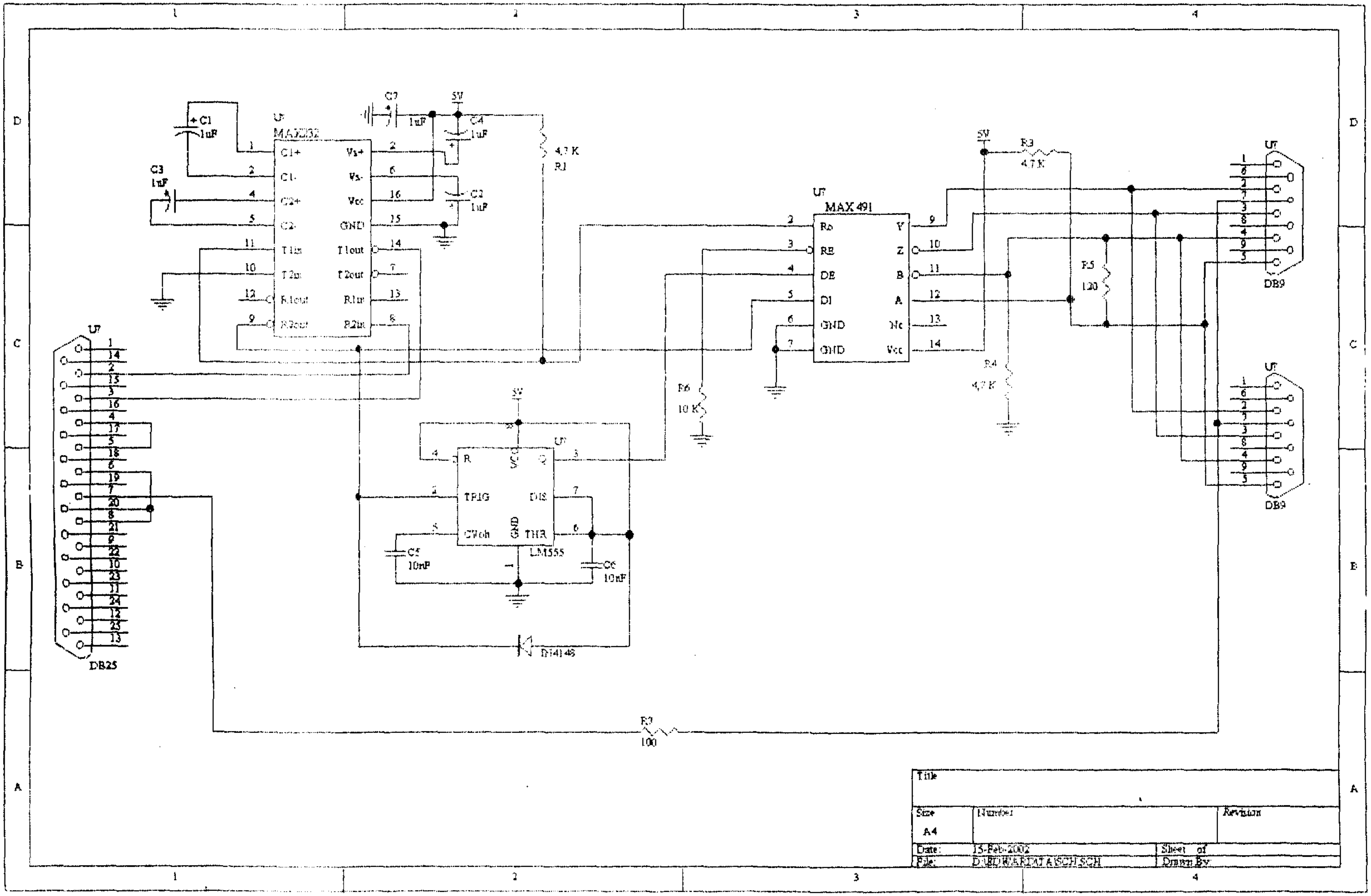
   Bila tidak sama maka master akan meminta slave untuk mengirim ulang

```
{MASTER}
unit main;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, ToolWin, Menus, ImgList, CPDrv, ExtCtrls, StdCtrls, jpeg;

type
  TForm1 = class(TForm)
    StatusBar: TStatusBar;
    ProgressBar1: TProgressBar;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Exit1: TMenuItem;
    Action1: TMenuItem;
    Connect1: TMenuItem;
    Capture1: TMenuItem;
    slave11: TMenuItem;
    slave21: TMenuItem;
    Checkslave1: TMenuItem;
    option1: TMenuItem;
    SelectCom1: TMenuItem;
    CommPortDriver: TCommPortDriver;
    ScrollBox1: TScrollBox;
    Image1: TImage;
    ToolBar2: TToolBar;
    conbtn: TToolButton;
    disconbtn: TToolButton;
    setbtn: TToolButton;
    checkbtn: TToolButton;
    ToolButton1: TToolButton;
    cap1btn: TToolButton;
    cap2btn: TToolButton;
    ToolButton4: TToolButton;
    exitbtn: TToolButton;
    aboutbtn: TToolButton;
    ImageList1: TImageList;
    Disconnect1: TMenuItem;
    About1: TMenuItem;
    procedure conbtnClick(Sender: TObject);
    procedure disconbtnClick(Sender: TObject);
    procedure setbtnClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure checkbtnClick(Sender: TObject);
    procedure CommPortDriverReceiveData(Sender: TObject; DataPtr: Pointer;
      DataSize: Cardinal);
    procedure FormCreate(Sender: TObject);
    procedure cap1btnClick(Sender: TObject);
    procedure putus;
    procedure aboutbtnClick(Sender: TObject);
    procedure Slave11Click(Sender: TObject);
    procedure exitbtnClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

 type mode=(none,service,besar,data);

 var
  Form1: TForm1;
  s,t:string;
  count,savecount:integer;
  slav1,slav2,cap1,cap2:boolean;
  work:mode;
  size:longint;
  repeated:integer;

implementation

uses selcom,aboutme;

{$R *.DFM}


procedure tform1.putus;
begin
 commportdriver.HwFlow:=hfnone;
 commportdriver.EnableDTROnOpen:=false;
 commportdriver.Disconnect;
end;

procedure delay(jum:integer);
begin
 repeat
  dec(jum);
  application.ProcessMessages;
 until jum=0;
end;



procedure TForm1.conbtnClick(Sender: TObject);
begin
  CommPortDriver.Parity:=ptSPACE;
  if commportdriver.connect then
  begin
    if commportdriver.getlinestatus=[] then
    begin
      commportdriver.checklinestatus:=true;
      commportdriver.HwFlow:=hfRTSCTS;
      commportdriver.EnableDTROnOpen:=true;
      conbtn.enabled:=false;
      connect1.enabled:=false;
      setbtn.enabled:=false;
      selectcom1.enabled:=false;
      disconbtn.enabled:=true;
      disconnect1.enabled:=true;
      statusbar.simpletext:='Connected to '+copy(commportdriver.PortName,length(comm
      checkbtn.enabled:=true;
      checkslave1.enabled:=true;
    end;
  end
  else
    begin
      statusbar.simpletext:='error:could not connect.check com port settings & try a
      messageBeep(0);
    end;
  end;
procedure TForm1.DisconbtnClick(Sender: TObject);
begin
  cap1btn.enabled:=false;
  cap2btn.enabled:=false;
  slave11.enabled:=false;
  slave21.enabled:=false;
  checkbtn.enabled:=false;
  checkslave1.enabled:=false;
  commportdriver.flushbuffers(true,false);
  progressBar1.position:=0;
  statusBar.simpletext:='Disconnected';
  putus;
  Disconbtn.enabled:=false;
  Disconnect1.enabled:=false;
```

```
    conbtn.enabled:=true;
    connect1.enabled:=true;
    setbtn.enabled:=true;
    selectcom1.enabled:=true;
  end;

procedure TForm1.setbtnClick(Sender: TObject);
begin
com:=tcom.Create(self);
com.ShowModal;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
 if commportdriver.Connected then
  begin
   commportdriver.flushbuffers(true,false);
   putus;
  end;
end;

procedure TForm1.checkbtnClick(Sender: TObject);
begin
cap1:=false;
cap2:=false;
commportdriver.SendString('#~@M1?');
delay(2000);
commportdriver.SendString('#~@M2?');
end;

procedure TForm1.CommPortDriverReceiveData(Sender: TObject;
  DataPtr: Pointer; DataSize: Cardinal);
var p:pchar;
    kode:char;
    f,tof:file;
    buf,temp:array[1..8186]of byte;
    tot,tulis,baca,frame,last:longint;
begin
work:=none;
frame:=0;
commportdriver.PausePolling;
s:='';
p:=dataptr;
t:='';
while datasize > 0 do
  begin
     if work=none then s:=s+p^;
     dec(datasize);
     if (length(s)>4) and (work=none) then t:=copy(s,length(s)-4,5);
     if work=service then kode:=p^;
     if work=besar then t:=t+p^;
     if work=data then
      begin
      if frame>0 then
       begin
        inc(count);
        buf[count]:=integer(p^);
        tot:=tot+buf[count];
       end
      else
       last:=integer(p^);
      dec(frame);
      end;
     if t='#~@S0' then
       work:=service;
     if (work=none) and (t='#~@D0') then
       begin
        work:=data;
        if size-savecount<8186 then
         frame:=size-savecount
        else
         frame:=8186;
        tot:=0;
        count:=0;
       end;
     if t='#~@B0' then
      begin
       work:=besar;
       t:='';
      end;
     inc(p);
  end;
if work=besar then
  begin
   size:=strtoint(t);
   repeated:=(size div 8186)+1;
   progressbar1.Position:=0;
   progressbar1.Max:=repeated;
   progressbar1.Visible:=true;
   savecount:=0;
   commportdriver.FlushBuffers(true,false);
   if cap1 then
     commportdriver.SendString('#~@M1=');
   if cap2 then
     commportdriver.SendString('#~@M2=');
  end;
if work=data then
  begin
  if (tot mod 256)<>last then
   begin
     if cap1 then commportdriver.SendString('#~@M1!');
     if cap2 then commportdriver.SendString('#~@M2!');
   end
  else
    begin
     dec(repeated);
     progressbar1.Position:=progressbar1.Position+1;
     if savecount=0 then
      begin
       assignfile(f,'c:\tmp.jpg');
       rewrite(f,1);
       blockwrite(f,buf,count);
       closefile(f);
      end
     else
      begin
       copyfile('c:\tmp.jpg','c:\tmp1.jpg',false);
       assignfile(f,'c:\tmp1.jpg');
       reset(f,1);
       assignfile(tof,'c:\tmp.jpg');
       rewrite(tof,1);
       repeat
        BlockRead(f, temp, SizeOf(temp), baca);
        BlockWrite(tof, temp, baca, tulis);
       until (baca = 0) or (tulis <> baca);
       seek(f,savecount);
       blockwrite(tof,buf,count);
       closefile(f);
       closefile(tof);
     end;
     savecount:=savecount+count;
//     memo1.Lines.Add(inttostr(savecount));
     if repeated=0 then
      begin
       image1.Picture.LoadFromFile('c:\tmp.jpg');
       progressbar1.Visible:=false;
       if cap1 then statusbar.SimpleText:='Capturing Image From Slave 1 Success';
       if cap2 then statusbar.SimpleText:='Capturing Image From Slave 2 Success';
       toolbar2.Enabled:=true;
       action1.Enabled:=true;
       option1.Enabled:=true;
       about1.Enabled:=true;
       deletefile('c:\temp1.jpg');
      end
     else
      begin
       if cap1 then
         commportdriver.SendString('#~@M1=');
       if cap2 then
         commportdriver.SendString('#~@M2=');
      end;
    end;
   commportdriver.FlushBuffers(true,false);
   work:=none;
  end;
if work=service then
 begin
  case kode of
   '1':begin
        slav1:=true;
       end;
   '2':begin
        slav2:=true;
       end;
   '3':begin
        slav1:=false;
       end;
   '4':begin
        slav2:=false;
       end;
  end;
  cap1btn.enabled:=slav1;
  slave11.enabled:=slav1;
  cap2btn.enabled:=slav2;
  slave21.enabled:=slav2;
 end;
commportdriver.ContinuePolling;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
slav1:=false;
slav2:=false;
work:=none;
end;

procedure TForm1.cap1btnClick(Sender: TObject);
begin
 count:='';
 cap1:=false;
 cap2:=false;
 statusbar.SimpleText:='Capturing Image in Process...Please Wait..!!!';
 case (sender as ttoolbutton).tag of
  1: begin
      commportdriver.SendString('#~@M1>');
      cap1:=true;
      end;
  2: begin
      commportdriver.SendString('#~@M2>');
      cap2:=true;
      end;
 end;
 toolbar2.Enabled:=false;
 action1.Enabled:=false;
 option1.Enabled:=false;
 about1.Enabled:=false;
end;

procedure TForm1.aboutbtnClick(Sender: TObject);
begin
close;
end;

procedure TForm1.Slave11Click(Sender: TObject);
begin
 count:=0;
 cap1:=false;
 cap2:=false;
 statusbar.SimpleText:='Capturing Image in Process...Please Wait..!!!';
 case (sender as tmenuitem).tag of
  1: begin
     commportdriver.SendString('#~@M1>');
     cap1:=true;
     end;
  2: begin
     commportdriver.SendString('#~@M2>');
     cap2:=true;
     end;
 end;
 toolbar2.Enabled:=false;
 action1.Enabled:=false;
 option1.Enabled:=false;

end;

procedure TForm1.exitbtnClick(Sender: TObject);
begin
aboutbox:=taboutbox.Create(self);
aboutbox.ShowModal;
end;

end.
```

```
{slave}
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, CPDrv, ComCtrls,jpeg, Menus,shellapi;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    ComboBox1: TComboBox;
    Button1: TButton;
    CommPortDriver: TCommPortDriver;
    PopupMenu1: TPopupMenu;
    HideMe1: TMenuItem;
    Connect1: TMenuItem;
    Exit1: TMenuItem;
    Label2: TLabel;
    ComboBox2: TComboBox;
    procedure Button1Click(Sender: TObject);
    procedure CommPortDriverReceiveData(Sender: TObject; DataPtr: Pointer;
      DataSize: Cardinal);
    procedure FormCreate(Sender: TObject);
    procedure putus;
    Procedure ulang;
    Procedure sendfile;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure HideMe1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure gantiicon;

   private
    procedure WndProc(var Msg : TMessage); override;
    { Private declarations }
   public
      IconData : TNotifyIconData;
      IconCount : integer
    { Public declarations }

   end;


var
  Form1: TForm1;
  s,t,posisi,perintah:string;
  work:boolean;
  size,count,savecount:longint;

implementation

{$R *.DFM}

procedure TForm1.WndProc(var Msg : TMessage);
var
  p : TPoint;
begin
  case Msg.Msg of
    WM_USER + 1:
    case Msg.lParam of
      WM_RBUTTONDOWN: begin
          GetCursorPos(p);
          PopupMenu1.Popup(p.x, p.y);
      end
    end;
  end;
  inherited;
```

```
end;


procedure tform1.putus;
begin
 perintah:='#~@S0'+inttostr(strtoint(posisi)+2);
 commportdriver.SendString(perintah);
 commportdriver.HwFlow:=hfnone;
 commportdriver.EnableDTROnOpen:=false;
 commportdriver.Disconnect;
end;


Procedure Tform1.sendfile;
var
  f:file;
  isi:array[1..8186]of byte;
  buf:array[1..8192]of byte;
  baca:integer;
  ada:boolean;
  a:integer;
  tot:longint;
begin
  commportdriver.flushbuffers(true,true);
  assignfile(f,'c:\tmp.jpg');
  {$I-} reset(f,1);
  {$I+};
  savecount:=count;
  seek(f,count);
  ada:=(IOresult=0);
  if ada then
    if commportdriver.connected then
     begin
      blockread(f,isi,sizeof(isi),baca);
      buf[1]:=integer('#');
      buf[2]:=integer('~');
      buf[3]:=integer('@');
      buf[4]:=integer('D');
      buf[5]:=integer('0');
      tot:=0;
      for a:=1 to baca do
       begin
        buf[5+a]:=isi[a];
        tot:=tot+isi[a];
       end;
      buf[6+baca]:=tot mod 256;
      commportdriver.senddata(@buf,6+baca);
      count:=count+baca;
     end;
  closefile(f);
end;

Procedure Tform1.ulang;
var
  f:file;
  isi:array[1..8186]of byte;
  buf:array[1..8192]of byte;
  baca:integer;
  ada:boolean;
  a:integer;
  tot:longint;
begin
  commportdriver.flushbuffers(true,true);
  assignfile(f,'c:\tmp.jpg');
  {$I-} reset(f,1);
  {$I+};
  count:=savecount;
  seek(f,count);
  ada:=(IOresult=0);
  if ada then
    if commportdriver.connected then
     begin
      blockread(f,isi,sizeof(isi),baca);
      buf[1]:=integer('#');
      buf[2]:=integer('~');
      buf[3]:=integer('@');
      buf[4]:=integer('D');
      buf[5]:=integer('0');
      tot:=0;
      for a:=1 to baca do
       begin
        buf[5+a]:=isi[a];
        tot:=tot+isi[a];
       end;
      buf[6+baca]:=tot mod 256;
      commportdriver.senddata(@buf,6+baca);
      count:=count+baca;
     end;
  closefile(f);
end;

procedure capture;
var gam:tbitmap;
    jp :TJpegImage;
    hand:hdc;
begin
  gam:=tbitmap.Create;
  hand:=getwindowdc(GetdesktopWindow);
  gam.Width:=screen.Width;
  gam.Height:=screen.Height;
  bitblt(gam.Canvas.Handle,0,0,gam.Width,gam.Height,hand,0,0,srccopy);
  jp := TJpegImage.Create;
  try
    with jp do
    begin
      Assign(gam);
      SaveToFile('c:\tmp.jpg');
    end;
  finally
    jp.Free;
  end;
  gam.Free;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
if (button1.Caption='Connect')or (connect1.Caption='Connect') then
 begin
  CommPortDriver.Parity:=ptSPACE;
   case ComboBox1.ItemIndex of
    0: Form1.commportdriver.PortName:='\\.\COM1';
    1: Form1.commportdriver.PortName:='\\.\COM2';
    else Form1.commportdriver.PortName:='\\.\COM1';
   end;
   case ComboBox2.ItemIndex of
    0: posisi:='1';
    1: posisi:='2';
    else posisi:='1';
   end;
  if commportdriver.connect then
  begin
    if commportdriver.getlinestatus=[] then
    begin
      commportdriver.HwFlow:=hfRTSCTS;
      commportdriver.EnableDTROnOpen:=true;
      commportdriver.checklinestatus:-true;
      perintah:='#~@S0'+posisi;
      commportdriver.SendString(perintah);
    end;
  end
  else
    begin
      application.MessageBox(pchar('Could not connect'+#13+'Try another port'),'Erro
      messageBeep(0);
    end;
 end
else
 putus;

if CommPortDriver.Connected then
 begin
   form1.Caption:='Slave '+posisi;
   button1.Caption:='Disconnect';
   connect1.Caption:='Disconnect';
 end
else
 begin
  button1.Caption:='Connect';
  connect1.Caption:='Connect';
 end;
gantiicon;
end;

procedure TForm1.CommPortDriverReceiveData(Sender: TObject;
  DataPtr: Pointer; DataSize: Cardinal);
var p:pchar;
    kode:char;
    f:file of byte;
    a:integer;
begin
commportdriver.PausePolling;
work:=false;
s:='';
p:=dataptr;
t:='';
while datasize > 0 do
  begin
     s:=s+p^;
     dec(datasize);
     if (length(s)>4) and (work=false) then t:=copy(s,length(s)-4,5);
     if work then kode:=p^;
     if t='#~@M'+posisi then
       work:=true;
     inc(p);
  end;
commportdriver.ContinuePolling;
case kode of
 '?':begin
      perintah:='#~@S0'+posisi;
      commportdriver.SendString(perintah);
     end;
 '>':begin
      capture;
      AssignFile(f,'c:\tmp.jpg');
      Reset(f);
      size:=filesize(f);
      s:=inttostr(size);
      closefile(f);
      count:=0;
      commportdriver.SendString('#~@B0'+s);
     end;
 '=':begin
      sendfile;
     end;
 '!':begin
     ulang;
    end;
end;
if ((kode='=') or (kode='!'))and(size>0)and (size<>count) then
 begin
  button1.Enabled:=false;
  for a:=0 to 2 do
   popupmenu1.Items[a].Enabled:=false;
 end;
if ((kode='=') or (kode='!'))and(size>0)and (size=count) then
 begin
  button1.Enabled:=true;
  for a:=0 to 2 do
   popupmenu1.Items[a].Enabled:=true;
 end;
end;


procedure tform1.gantiicon;
begin
if commportdriver.Connected then
 IconData.hIcon := Application.Icon.Handle
else
 IconData.hIcon := LoadIcon(0, IDI_HAND);
Application.Title := TimeToStr(Now);
StrPCopy(IconData.szTip, Application.Title);
Shell_NotifyIcon(NIM_MODIFY, @IconData);
end;


procedure TForm1.FormCreate(Sender: TObject);
begin
  work:=false;
  BorderIcons := [biSystemMenu];
  IconCount :=2;
  IconData.cbSize := sizeof(IconData);
  IconData.Wnd := Handle;
  IconData.uID := 100;
  IconData.uFlags := NIF_MESSAGE + NIF_ICON + NIF_TIP;
  IconData.uCallbackMessage := WM_USER + 1;
  IconData.hIcon := Application.Icon.Handle;
  StrPCopy(IconData.szTip, Application.Title);
  Shell_NotifyIcon(NIM_ADD, @IconData);
  Form1.Show;
  ShowWindow(Application.Handle, SW_HIDE);
  posisi:='1';
  gantiicon;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Action := caNone;
Form1.Hide;
hideme1.Caption:='&Show Me';
end;

procedure TForm1.HideMe1Click(Sender: TObject);
begin
 if hideme1.Caption='&Hide Me' then
  begin
   close;
  end
 else
 begin
   Form1.Show;
   ShowWindow(Application.Handle, SW_HIDE);
   hideme1.Caption:='&Hide Me';
   exit;
  end
end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
putus;
Shell_NotifyIcon(NIM_DELETE, @IconData);
Application.ProcessMessages;
Application.Terminate;
end;

end.
```



U?
MAX232
C1+
C1-
C2+
C2-
T1in
T2in
R1out
R2out
Vs+
Vs-
Vcc
GND
T1out
T2out
R1in
R2in
C1
1uF
C2
1uF
C3
1uF
C4
1uF
C7
1uF
5V
4.7 K
R1
U?
MAX 491
Ro
RE
DE
DI
GND
GND
Y
Z
B
A
Nc
Vcc
R3
4.7 K
R4
4.7 K
R5
120
R6
10 K
R7
100
U?
LM555
R
Vcc
Q
TRIG
DIS
CVolt
GND
THR
C5
10nF
C6
10nF
D1N4148
U?
DB25
U?
DB9
U?
DB9
1
2
3
4
A
B
C
D
Title
Size
A4
Number
Revision
Date:
15-Feb-2002
Sheet of
File:
D:\BDWARTATA\SCH.SCH
Drawn By:

LAMPIRAN

19-0122; Rev 5; 2/96

MAXIM

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

General Description

The MAX481, MAX483, MAX485, MAX487–MAX491, and MAX1487 are low-power transceivers for RS-485 and RS-422 communication. Each part contains one driver and one receiver. The MAX483, MAX487, MAX488, and MAX489 feature reduced slew-rate drivers that minimize EMI and reduce reflections caused by improperly terminated cables, thus allowing error-free data transmission up to 250kbps. The driver slew rates of the MAX481, MAX485, MAX490, MAX491, and MAX1487 are not limited, allowing them to transmit up to 2.5Mbps.

These transceivers draw between 120µA and 500µA of supply current when unloaded or fully loaded with disabled drivers. Additionally, the MAX481, MAX483, and MAX487 have a low-current shutdown mode in which they consume only 0.1µA. All parts operate from a single 5V supply.

Drivers are short-circuit current limited and are protected against excessive power dissipation by thermal shutdown circuitry that places the driver outputs into a high-impedance state. The receiver input has a fail-safe feature that guarantees a logic-high output if the input is open circuit.

The MAX487 and MAX1487 feature quarter-unit-load receiver input impedance, allowing up to 128 MAX487/MAX1487 transceivers on the bus. Full-duplex communications are obtained using the MAX488–MAX491, while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are designed for half-duplex applications.

Applications

Low-Power RS-485 Transceivers
Low-Power RS-422 Transceivers
Level Translators
Transceivers for EMI-Sensitive Applications
Industrial-Control Local Area Networks

Features

- ♦ **In µMAX Package: Smallest 8-Pin SO**
- ♦ **Slew-Rate Limited for Error-Free Data Transmission (MAX483/487/488/489)**
- ♦ **0.1µA Low-Current Shutdown Mode (MAX481/483/487)**
- ♦ **Low Quiescent Current:**
  120µA (MAX483/487/488/489)
  230µA (MAX1487)
  300µA (MAX481/485/490/491)
- ♦ **-7V to +12V Common-Mode Input Voltage Range**
- ♦ **Three-State Outputs**
- ♦ **30ns Propagation Delays, 5ns Skew (MAX481/485/490/491/1487)**
- ♦ **Full-Duplex and Half-Duplex Versions Available**
- ♦ **Operate from a Single 5V Supply**
- ♦ **Allows up to 128 Transceivers on the Bus (MAX487/MAX1487)**
- ♦ **Current-Limiting and Thermal Shutdown for Driver Overload Protection**

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX481CPA	0°C to +70°C	8 Plastic DIP
MAX481CSA	0°C to +70°C	8 SO
MAX481CUA	0°C to +70°C	8 µMAX
MAX481C/D	0°C to +70°C	Dice*

Ordering Information continued at end of data sheet.

** Contact factory for dice specifications.*

Selection Table

PART NUMBER	HALF/FULL DUPLEX	DATA RATE (Mbps)	SLEW-RATE LIMITED	LOW-POWER SHUTDOWN	RECEIVER/ DRIVER ENABLE	QUIESCENT CURRENT (µA)	NUMBER OF TRANSMITTERS ON BUS	PIN COUNT
MAX481	Half	2.5	No	Yes	Yes	300	32	8
MAX483	Half	0.25	Yes	Yes	Yes	120	32	8
MAX485	Half	2.5	No	No	Yes	300	32	8
MAX487	Half	0.25	Yes	Yes	Yes	120	128	8
MAX488	Full	0.25	Yes	No	No	120	32	8
MAX489	Full	0.25	Yes	No	Yes	120	32	14
MAX490	Full	2.5	No	No	No	300	32	8
MAX491	Full	2.5	No	No	Yes	300	32	14
MAX1487	Half	2.5	No	No	Yes	230	128	8

ABSOLUTE MAXIMUM RATINGS

Supply Voltage (V_{CC})	12V
Control Input Voltage ($\overline{RE}$, DE)	-0.5V to (V_{CC} + 0.5V)
Driver Input Voltage (DI)	-0.5V to (V_{CC} + 0.5V)
Driver Output Voltage (A, B)	-8V to +12.5V
Receiver Input Voltage (A, B)	-8V to +12.5V
Receiver Output Voltage (RO)	-0.5V to (V_{CC} +0.5V)
Continuous Power Dissipation (T_A = +70°C)	
8-Pin Plastic DIP (derate 9.09mW/°C above +70°C)	727mW
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C)	800mW
8-Pin SO (derate 5.88mW/°C above +70°C)	471mW
14-Pin SO (derate 8.33mW/°C above +70°C)	667mW
8-Pin µMAX (derate 4.1mW/°C above +70°C)	830mW
8-Pin CERDIP (derate 8.00mW/°C above +70°C)	640mW
14-Pin CERDIP (derate 9.09mW/°C above +70°C)	727mW
Operating Temperature Ranges	
MAX4_ _C_ _/MAX1487C_ A	0°C to +70°C
MAX4_ _E_ _/MAX1487E_ A	-40°C to +85°C
MAX4_ _MJ_/MAX1487MJA	-55°C to +125°C
Storage Temperature Range	-65°C to +160°C
Lead Temperature (soldering, 10sec)	+300°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 in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Differential Driver Output (no load)	V_{OD1}					5	V
Differential Driver Output (with load)	V_{OD2}	R = 50Ω (RS-422)		2			V
		R = 27Ω (RS-485), Figure 4		1.5		5	
Change in Magnitude of Driver Differential Output Voltage for Complementary Output States	ΔV_{OD}	R = 27Ω or 50Ω, Figure 4				0.2	V
Driver Common-Mode Output Voltage	V_{OC}	R = 27Ω or 50Ω, Figure 4				3	V
Change in Magnitude of Driver Common-Mode Output Voltage for Complementary Output States	ΔV_{OD}	R = 27Ω or 50Ω, Figure 4				0.2	V
Input High Voltage	V_{IH}	DE, DI, $\overline{RE}$		2.0			V
Input Low Voltage	V_{IL}	DE, DI, $\overline{RE}$				0.8	V
Input Current	I_{IN1}	DE, DI, $\overline{RE}$				±2	µA
Input Current (A, B)	I_{IN2}	DE = 0V; V_{CC} = 0V or 5.25V, all devices except MAX487/MAX1487	V_{IN} = 12V			1.0	mA
			V_{IN} = -7V			-0.8	
		MAX487/MAX1487, DE = 0V, V_{CC} = 0V or 5.25V	V_{IN} = 12V			0.25	mA
			V_{IN} = -7V			-0.2	
Receiver Differential Threshold Voltage	V_{TH}	-7V ≤ V_{CM} ≤ 12V		-0.2		0.2	V
Receiver Input Hysteresis	ΔV_{TH}	V_{CM} = 0V			70		mV
Receiver Output High Voltage	V_{OH}	I_O = -4mA, V_{ID} = 200mV		3.5			V
Receiver Output Low Voltage	V_{OL}	I_O = 4mA, V_{ID} = -200mV				0.4	V
Three-State (high impedance) Output Current at Receiver	I_{OZR}	0.4V ≤ V_O ≤ 2.4V				±1	µA
Receiver Input Resistance	R_{IN}	-7V ≤ V_{CM} ≤ 12V, all devices except MAX487/MAX1487		12			kΩ
		-7V ≤ V_{CM} ≤ 12V, MAX487/MAX1487		48			kΩ

DC ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS			MIN	TYP	MAX	UNITS
No-Load Supply Current (Note 3)	I_{CC}	MAX488/MAX489, DE, DI, $\overline{RE}$ = 0V or V_{CC}				120	250	μA
		MAX490/MAX491, DE, DI, $\overline{RE}$ = 0V or V_{CC}				300	500	
		MAX481/MAX485, $\overline{RE}$ = 0V or V_{CC}	DE = V_{CC}			500	900	
			DE = 0V			300	500	
		MAX1487, $\overline{RE}$ = 0V or V_{CC}	DE = V_{CC}			300	500	
			DE = 0V			230	400	
		MAX483/MAX487, $\overline{RE}$ = 0V or V_{CC}	DE = 5V	MAX483		350	650	
				MAX487		250	400	
			DE = 0V			120	250	
Supply Current in Shutdown	I_{SHDN}	MAX481/483/487, DE = 0V, $\overline{RE}$ = V_{CC}				0.1	10	μA
Driver Short-Circuit Current, V_O = High	I_{OSD1}	-7V ≤ V_O ≤12V (Note 4)			35		250	mA
Driver Short-Circuit Current, V_O = Low	I_{OSD2}	-7V ≤ V_O ≤12V (Note 4)			35		250	mA
Receiver Short-Circuit Current	I_{OSR}	0V ≤ V_O ≤ V_{CC}			7		95	mA

SWITCHING CHARACTERISTICS—MAX481/MAX485, MAX490/MAX491, MAX1487

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Driver Input to Output	t_{PLH}	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF		10	30	60	ns
	t_{PHL}			10	30	60	
Driver Output Skew to Output	t_{SKEW}	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF			5	10	ns
Driver Rise or Fall Time	t_R, t_F	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	MAX481, MAX485, MAX1487	3	15	40	ns
			MAX490C/E, MAX491C/E	5	15	25	
			MAX490M, MAX491M	3	15	40	
Driver Enable to Output High	t_{ZH}	Figures 7 and 9, C_L = 100pF, S2 closed			40	70	ns
Driver Enable to Output Low	t_{ZL}	Figures 7 and 9, C_L = 100pF, S1 closed			40	70	ns
Driver Disable Time from Low	t_{LZ}	Figures 7 and 9, C_L = 15pF, S1 closed			40	70	ns
Driver Disable Time from High	t_{HZ}	Figures 7 and 9, C_L = 15pF, S2 closed			40	70	ns
Receiver Input to Output	t_{PLH}, t_{PHL}	Figures 6 and 10, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	MAX481, MAX485, MAX1487	20	90	200	ns
			MAX490C/E, MAX491C/E	20	90	150	
			MAX490M, MAX491M	20	90	200	
| t_{PLH} - t_{PHL} | Differential Receiver Skew	t_{SKD}	Figures 6 and 10, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF			13		ns
Receiver Enable to Output Low	t_{ZL}	Figures 5 and 11, C_{RL} = 15pF, S1 closed			20	50	ns
Receiver Enable to Output High	t_{ZH}	Figures 5 and 11, C_{RL} = 15pF, S2 closed			20	50	ns
Receiver Disable Time from Low	t_{LZ}	Figures 5 and 11, C_{RL} = 15pF, S1 closed			20	50	ns
Receiver Disable Time from High	t_{HZ}	Figures 5 and 11, C_{RL} = 15pF, S2 closed			20	50	ns
Maximum Data Rate	f_{MAX}			2.5			Mbps
Time to Shutdown	t_{SHDN}	MAX481 (Note 5)		50	200	600	ns

SWITCHING CHARACTERISTICS—MAX481/MAX485, MAX490/MAX491, MAX1487 (continued)

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Driver Enable from Shutdown to Output High (MAX481)	$t_{ZH(SHDN)}$	Figures 7 and 9, C_L = 100pF, S2 closed		40	100	ns
Driver Enable from Shutdown to Output Low (MAX481)	$t_{ZL(SHDN)}$	Figures 7 and 9, C_L = 100pF, S1 closed		40	100	ns
Receiver Enable from Shutdown to Output High (MAX481)	$t_{ZH(SHDN)}$	Figures 5 and 11, C_L = 15pF, S2 closed, A - B = 2V		300	1000	ns
Receiver Enable from Shutdown to Output Low (MAX481)	$t_{ZL(SHDN)}$	Figures 5 and 11, C_L = 15pF, S1 closed, B - A = 2V		300	1000	ns

SWITCHING CHARACTERISTICS—MAX483, MAX487/MAX488/MAX489

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Driver Input to Output	t_{PLH}	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	250	800	2000	ns
	t_{PHL}		250	800	2000	
Driver Output Skew to Output	t_{SKEW}	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF		100	800	ns
Driver Rise or Fall Time	t_R, t_F	Figures 6 and 8, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	250		2000	ns
Driver Enable to Output High	t_{ZH}	Figures 7 and 9, C_L = 100pF, S2 closed	250		2000	ns
Driver Enable to Output Low	t_{ZL}	Figures 7 and 9, C_L = 100pF, S1 closed	250		2000	ns
Driver Disable Time from Low	t_{LZ}	Figures 7 and 9, C_L = 15pF, S1 closed	300		3000	ns
Driver Disable Time from High	t_{HZ}	Figures 7 and 9, C_L = 15pF, S2 closed	300		3000	ns
Receiver Input to Output	t_{PLH}	Figures 6 and 10, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF	250		2000	ns
	t_{PHL}		250		2000	
$\mid t_{PLH} - t_{PHL} \mid$ Differential Receiver Skew	t_{SKD}	Figures 6 and 10, R_{DIFF} = 54Ω, C_{L1} = C_{L2} = 100pF		100		ns
Receiver Enable to Output Low	t_{ZL}	Figures 5 and 11, C_{RL} = 15pF, S1 closed		20	50	ns
Receiver Enable to Output High	t_{ZH}	Figures 5 and 11, C_{RL} = 15pF, S2 closed		20	50	ns
Receiver Disable Time from Low	t_{LZ}	Figures 5 and 11, C_{RL} = 15pF, S1 closed		20	50	ns
Receiver Disable Time from High	t_{HZ}	Figures 5 and 11, C_{RL} = 15pF, S2 closed		20	50	ns
Maximum Data Rate	f_{MAX}	t_{PLH}, t_{PHL} < 50% of data period	250			kbps
Time to Shutdown	t_{SHDN}	MAX483/MAX487 (Note 5)	50	200	600	ns
Driver Enable from Shutdown to Output High	$t_{ZH(SHDN)}$	MAX483/MAX487, Figures 7 and 9, C_L = 100pF, S2 closed			2000	ns
Driver Enable from Shutdown to Output Low	$t_{ZL(SHDN)}$	MAX483/MAX487, Figures 7 and 9, C_L = 100pF, S1 closed			2000	ns
Receiver Enable from Shutdown to Output High	$t_{ZH(SHDN)}$	MAX483/MAX487, Figures 5 and 11, C_L = 15pF, S2 closed			2500	ns
Receiver Enable from Shutdown to Output Low	$t_{ZL(SHDN)}$	MAX483/MAX487, Figures 5 and 11, C_L = 15pF, S1 closed			2500	ns

NOTES FOR ELECTRICAL/SWITCHING CHARACTERISTICS

Note 1: All currents into device pins are positive; all currents out of device pins are negative. All voltages are referenced to device ground unless otherwise specified.

Note 2: All typical specifications are given for V_{CC} = 5V and T_A = +25°C.

Note 3: Supply current specification is valid for loaded transmitters when DE = 0V.

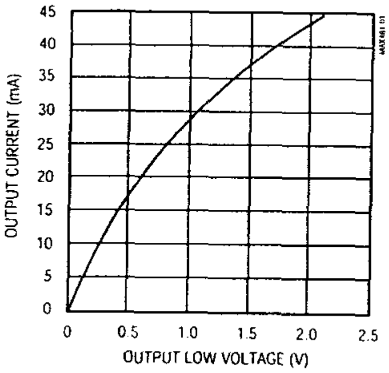
Note 4: Applies to peak current. See *Typical Operating Characteristics.*

Note 5: The MAX481/MAX483/MAX487 are put into shutdown by bringing $\overline{RE}$ high and DE low. If the inputs are in this state for less than 50ns, the parts are guaranteed not to enter shutdown. If the inputs are in this state for at least 600ns, the parts are guaranteed to have entered shutdown. See *Low-Power Shutdown Mode* section.

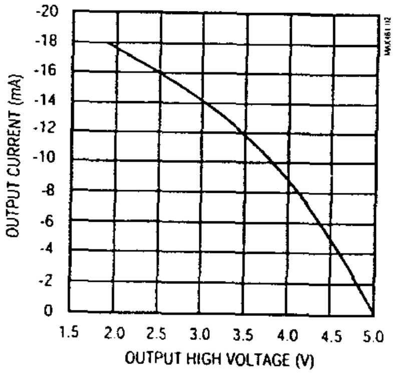
Typical Operating Characteristics

(V_{CC} = 5V, T_A = +25°C, unless otherwise noted.)

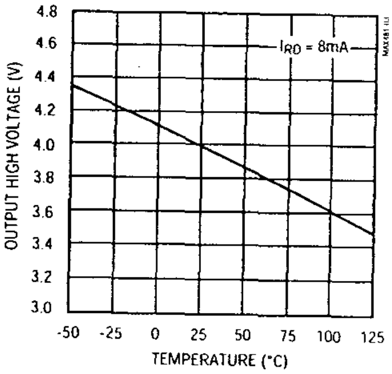
OUTPUT CURRENT vs. RECEIVER OUTPUT LOW VOLTAGE



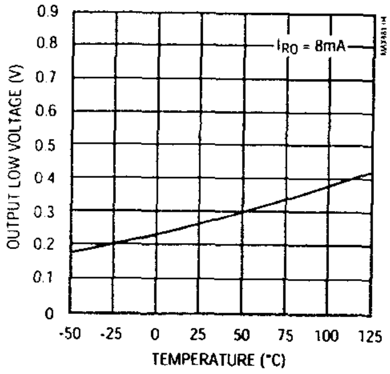
OUTPUT CURRENT vs. RECEIVER OUTPUT HIGH VOLTAGE



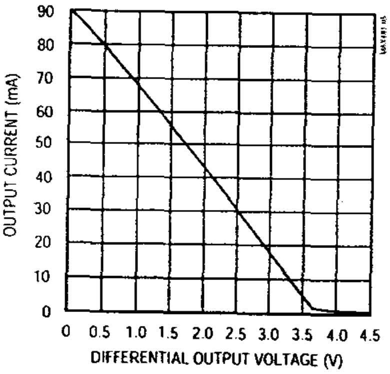
RECEIVER OUTPUT HIGH VOLTAGE vs. TEMPERATURE



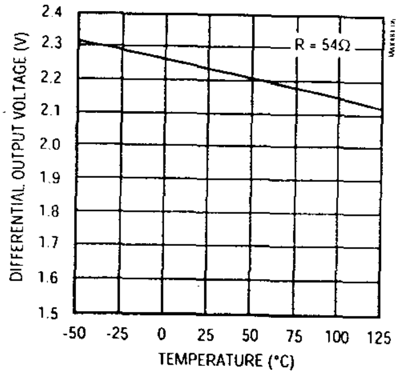
RECEIVER OUTPUT LOW VOLTAGE vs. TEMPERATURE



DRIVER OUTPUT CURRENT vs. DIFFERENTIAL OUTPUT VOLTAGE



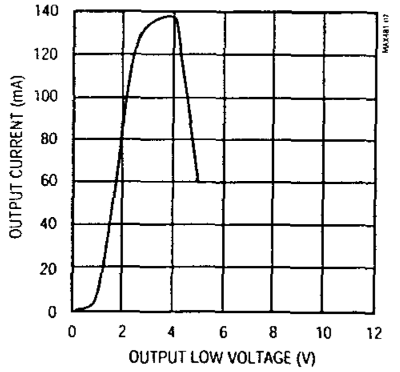
DRIVER DIFFERENTIAL OUTPUT VOLTAGE vs. TEMPERATURE



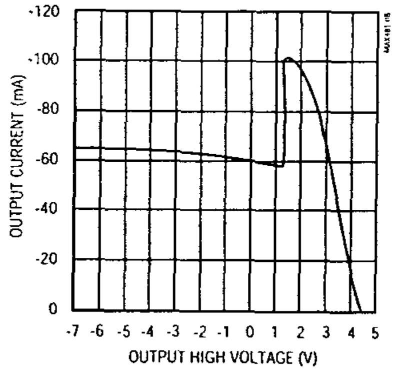
Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Typical Operating Characteristics (continued)

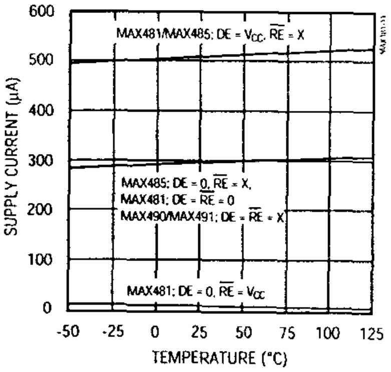
(V_{CC} = 5V, T_A = +25°C, unless otherwise noted.)



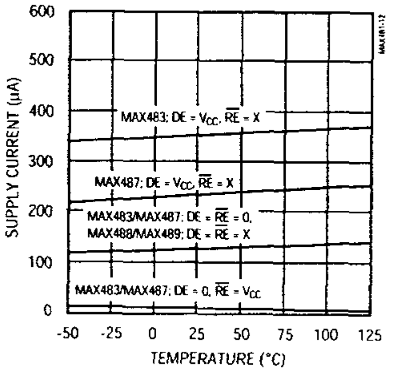
OUTPUT CURRENT vs. DRIVER OUTPUT LOW VOLTAGE



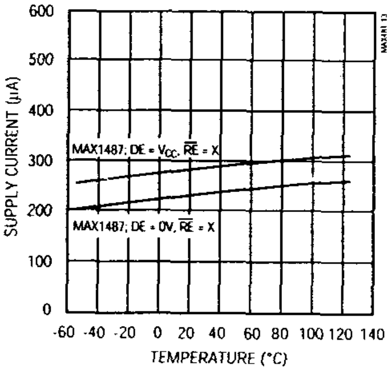
OUTPUT CURRENT vs. DRIVER OUTPUT HIGH VOLTAGE



MAX481/MAX485/MAX490/MAX491 SUPPLY CURRENT vs. TEMPERATURE



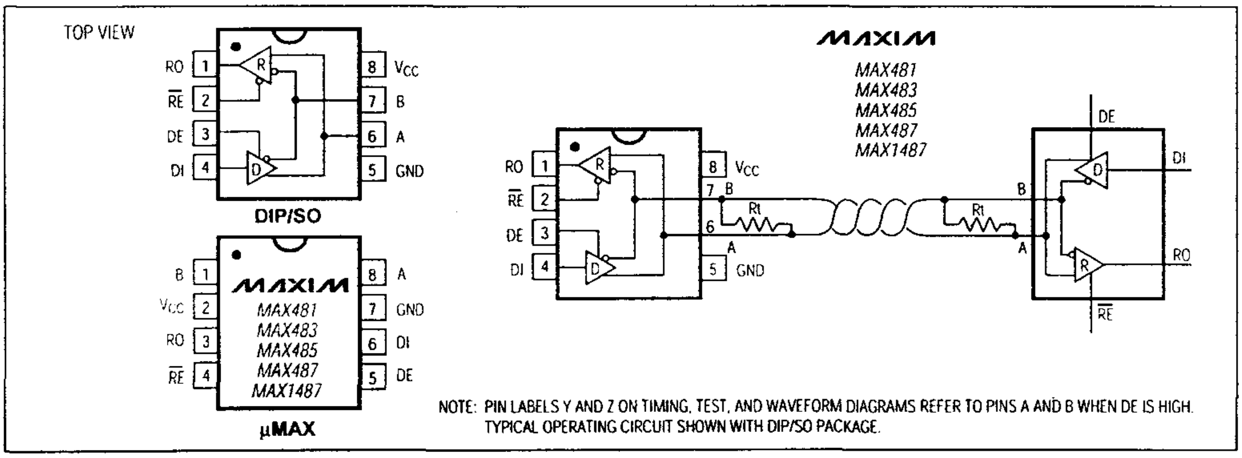
MAX483/MAX487–MAX489 SUPPLY CURRENT vs. TEMPERATURE



MAX1487 SUPPLY CURRENT vs. TEMPERATURE

Pin Description

PIN					NAME	FUNCTION
MAX481/MAX483/ MAX485/MAX487/ MAX1487		MAX488/ MAX490		MAX489/ MAX491		
DIP/SO	μMAX	DIP/SO	μMAX	DIP/SO		
1	3	2	4	2	RO	Receiver Output: If A > B by 200mV, RO will be high; If A < B by 200mV, RO will be low.
2	4	—	—	3	$\overline{RE}$	Receiver Output Enable. RO is enabled when $\overline{RE}$ is low; RO is high impedance when $\overline{RE}$ is high.
3	5	—	—	4	DE	Driver Output Enable. The driver outputs, Y and Z, are enabled by bringing DE high. They are high impedance when DE is low. If the driver outputs are enabled, the parts function as line drivers. While they are high impedance, they function as line receivers if $\overline{RE}$ is low.
4	6	3	5	5	DI	Driver Input. A low on DI forces output Y low and output Z high. Similarly, a high on DI forces output Y high and output Z low.
5	7	4	6	6, 7	GND	Ground
—	—	5	7	9	Y	Noninverting Driver Output
—	—	6	8	10	Z	Inverting Driver Output
6	8	—	—	—	A	Noninverting Receiver Input and Noninverting Driver Output
—	—	8	2	12	A	Noninverting Receiver Input
7	1	—	—	—	B	Inverting Receiver Input and Inverting Driver Output
—	—	7	1	11	B	Inverting Receiver Input
8	2	1	3	14	V_{CC}	Positive Supply: 4.75V ≤ V_{CC} ≤ 5.25V
—	—	—	—	1, 8, 13	N.C.	No Connect—not internally connected



NOTE: PIN LABELS Y AND Z ON TIMING, TEST, AND WAVEFORM DIAGRAMS REFER TO PINS A AND B WHEN DE IS HIGH.
TYPICAL OPERATING CIRCUIT SHOWN WITH DIP/SO PACKAGE.

Figure 1. MAX481/MAX483/MAX485/MAX487/MAX1487 Pin Configuration and Typical Operating Circuit

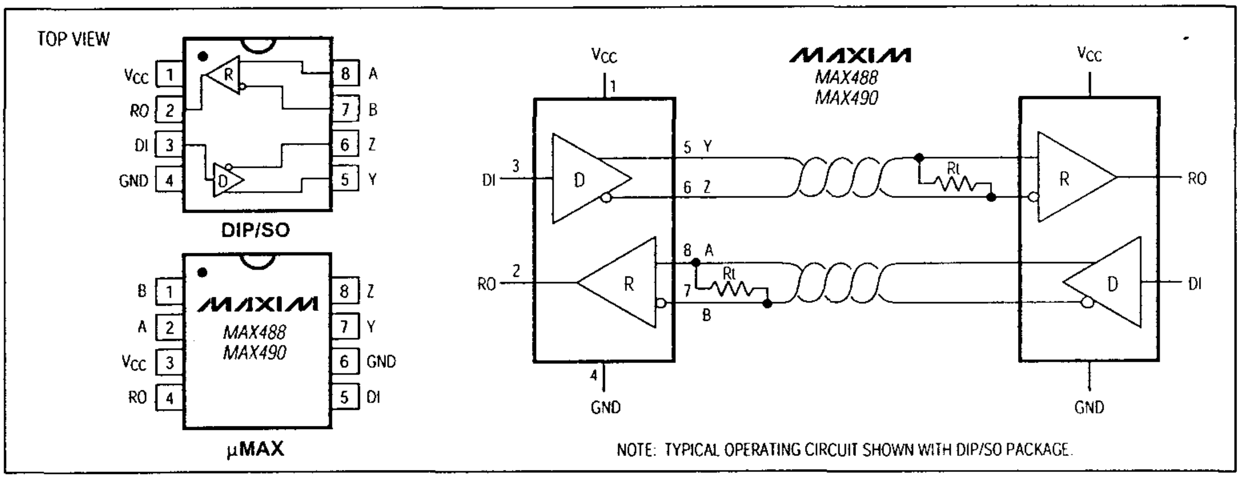


Figure 2. MAX488/MAX490 Pin Configuration and Typical Operating Circuit

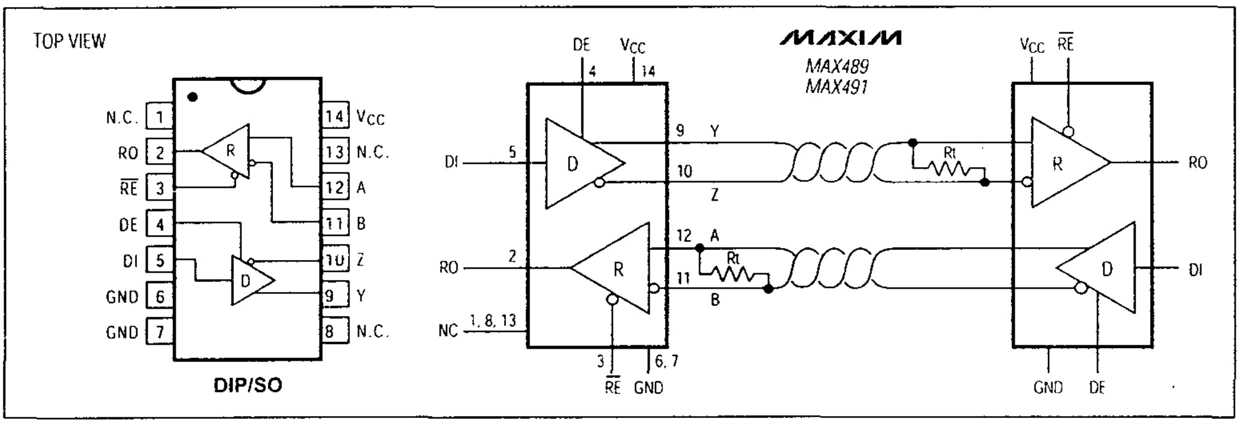


Figure 3. MAX489/MAX491 Pin Configuration and Typical Operating Circuit

Applications Information

The MAX481/MAX483/MAX485/MAX487-MAX491 and MAX1487 are low-power transceivers for RS-485 and RS-422 communications. The MAX481, MAX485, MAX490, MAX491, and MAX1487 can transmit and receive at data rates up to 2.5Mbps, while the MAX483, MAX487, MAX488, and MAX489 are specified for data rates up to 250kbps. The MAX488-MAX491 are full-duplex transceivers while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are half-duplex. In addition, Driver Enable (DE) and Receiver Enable ($\overline{RE}$) pins are included on the MAX481, MAX483, MAX485, MAX487, MAX489, MAX491, and MAX1487. When disabled, the driver and receiver outputs are high impedance.

MAX487/MAX1487: 128 Transceivers on the Bus

The 48kΩ, $^1/_4$-unit-load receiver input impedance of the MAX487 and MAX1487 allows up to 128 transceivers on a bus, compared to the 1-unit load (12kΩ input impedance) of standard RS-485 drivers (32 transceivers maximum). Any combination of MAX487/MAX1487 and other RS-485 transceivers with a total of 32 unit loads or less can be put on the bus. The MAX481/MAX483/MAX485 and MAX488-MAX491 have standard 12kΩ Receiver Input impedance.

Test Circuits

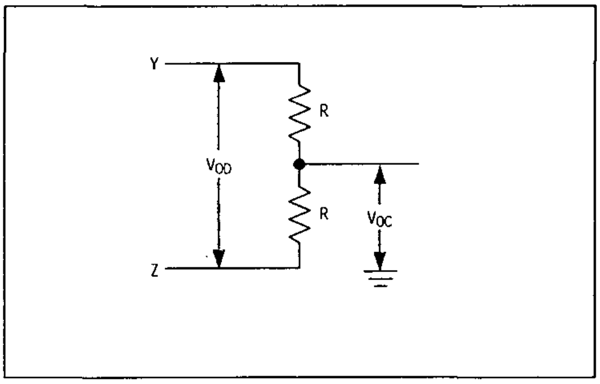


Figure 4. Driver DC Test Load

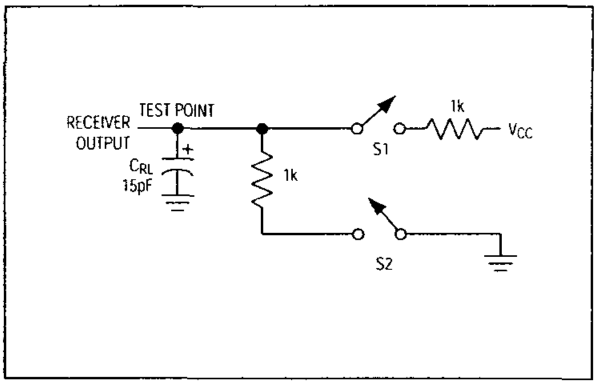


Figure 5. Receiver Timing Test Load

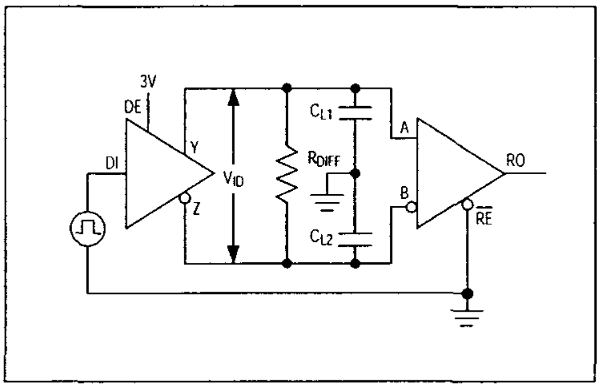


Figure 6. Driver/Receiver Timing Test Circuit

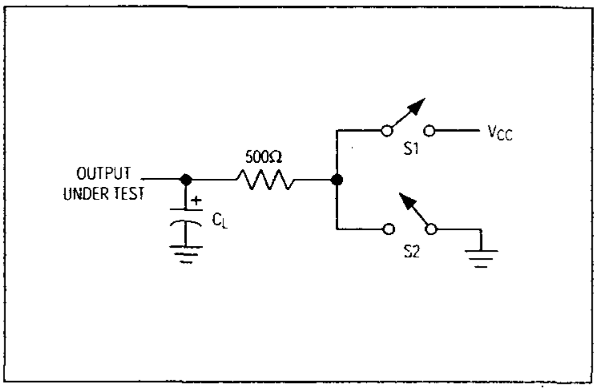


Figure 7. Driver Timing Test Load

MAX483/MAX487/MAX488/MAX489: Reduced EMI and Reflections

The MAX483 and MAX487–MAX489 are slew-rate limited, minimizing EMI and reducing reflections caused by improperly terminated cables. Figure 12 shows the driver output waveform and its Fourier analysis of a 150kHz signal transmitted by a MAX481, MAX485, MAX490, MAX491, or MAX1487. High-frequency harmonics with large amplitudes are evident. Figure 13 shows the same information displayed for a MAX483, MAX487, MAX488, or MAX489 transmitting under the same conditions. Figure 13's high-frequency harmonics have much lower amplitudes, and the potential for EMI is significantly reduced.

Switching Waveforms

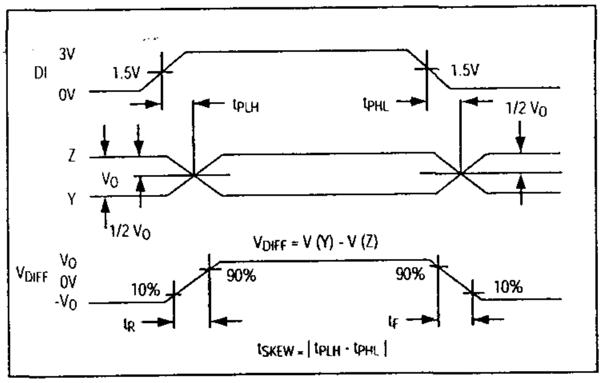


Figure 8. Driver Propagation Delays

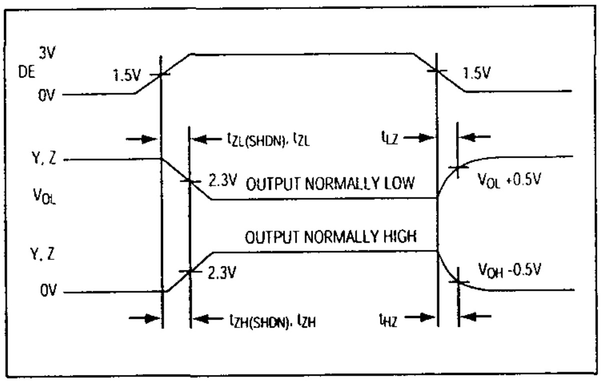


Figure 9. Driver Enable and Disable Times (except MAX488 and MAX490)

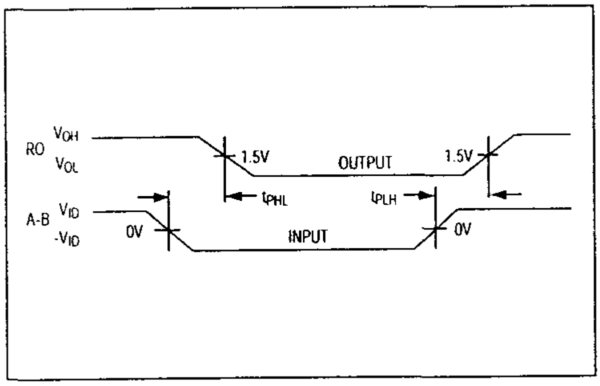


Figure 10. Receiver Propagation Delays

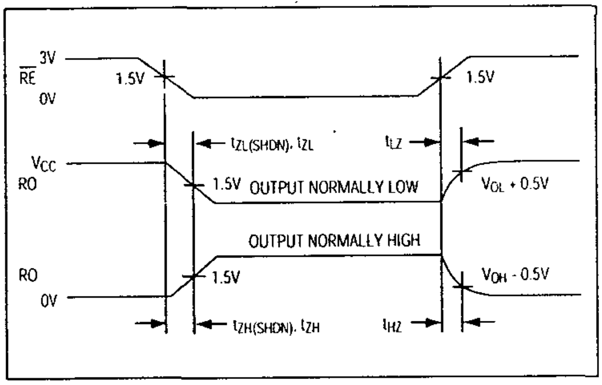


Figure 11. Receiver Enable and Disable Times (except MAX488 and MAX490)

Function Tables (MAX481/MAX483/MAX485/MAX487/MAX1487)

Table 1. Transmitting

INPUTS			OUTPUTS	
$\overline{RE}$	DE	DI	Z	Y
X	1	1	0	1
X	1	0	1	0
0	0	X	High-Z	High-Z
1	0	X	High-Z*	High-Z*

X = Don't care
High-Z = High impedance
* Shutdown mode for MAX481/MAX483/MAX487

Table 2. Receiving

INPUTS			OUTPUT
$\overline{RE}$	DE	A-B	RO
0	0	≥ +0.2V	1
0	0	≤ -0.2V	0
0	0	Inputs open	1
1	0	X	High-Z*

X = Don't care
High-Z = High impedance
* Shutdown mode for MAX481/MAX483/MAX487

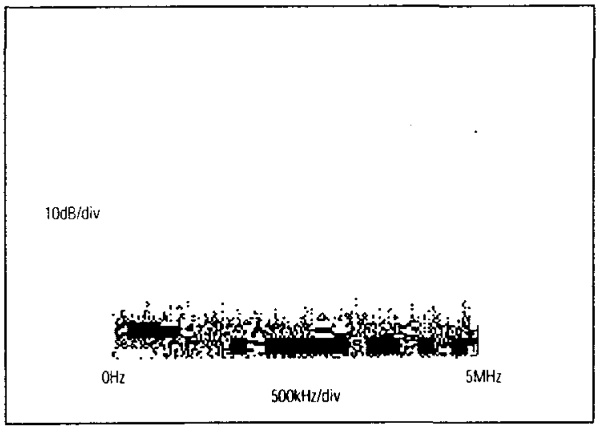


Figure 12. Driver Output Waveform and FFT Plot of MAX481/ MAX485/MAX490/MAX491/MAX1487 Transmitting a 150kHz Signal

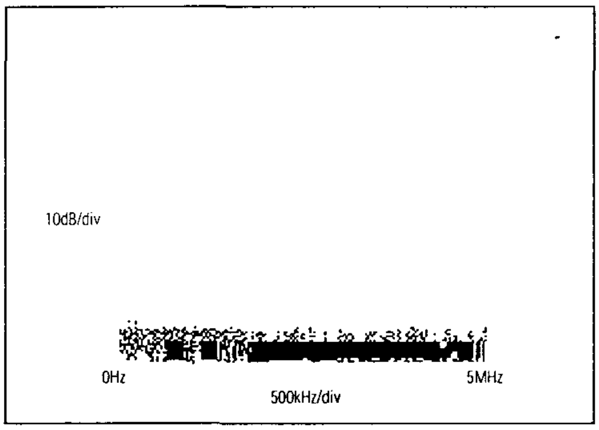


Figure 13. Driver Output Waveform and FFT Plot of MAX483/ MAX487–MAX489 Transmitting a 150kHz Signal

Low-Power Shutdown Mode (MAX481/MAX483/MAX487)

A low-power shutdown mode is initiated by bringing both $\overline{RE}$ high and DE low. The devices will not shut down unless both the driver and receiver are disabled. In shutdown, the devices typically draw only 0.1µA of supply current.

$\overline{RE}$ and DE may be driven simultaneously; the parts are guaranteed not to enter shutdown if $\overline{RE}$ is high and DE is low for less than 50ns. If the inputs are in this state for at least 600ns, the parts are guaranteed to enter shutdown.

For the MAX481, MAX483, and MAX487, the t_{ZH} and t_{ZL} enable times assume the part was not in the low-power shutdown state (the MAX485/MAX488–MAX491 and MAX1487 can not be shut down). The $t_{ZH(SHDN)}$ and $t_{ZL(SHDN)}$ enable times assume the parts were shut down (see *Electrical Characteristics*).

It takes the drivers and receivers longer to become enabled from the low-power shutdown state ($t_{ZH(SHDN)}$, $t_{ZL(SHDN)}$) than from the operating mode (t_{ZH}, t_{ZL}). (The parts are in operating mode if the $\overline{RE}$, DE inputs equal a logical 0,1 or 1,1 or 0, 0.)

Driver Output Protection

Excessive output current and power dissipation caused by faults or by bus contention are prevented by two mechanisms. A foldback current limit on the output stage provides immediate protection against short circuits over the whole common-mode voltage range (see *Typical Operating Characteristics*). In addition, a thermal shutdown circuit forces the driver outputs into a high-impedance state if the die temperature rises excessively.

Propagation Delay

Many digital encoding schemes depend on the difference between the driver and receiver propagation delay times. Typical propagation delays are shown in Figures 15–18 using Figure 14's test circuit.

The difference in receiver delay times, $| t_{PLH} - t_{PHL} |$, is typically under 13ns for the MAX481, MAX485, MAX490, MAX491, and MAX1487 and is typically less than 100ns for the MAX483 and MAX487–MAX489.

The driver skew times are typically 5ns (10ns max) for the MAX481, MAX485, MAX490, MAX491, and MAX1487, and are typically 100ns (800ns max) for the MAX483 and MAX487–MAX489.

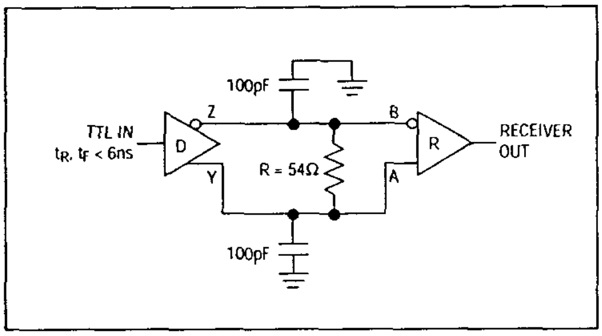


Figure 14. Receiver Propagaticn Delay Test Circuit

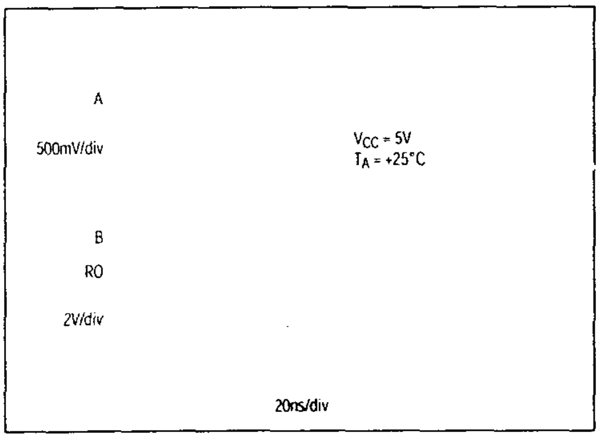


Figure 15. MAX481/MAX485/MAX490/MAX491/MAX1487 Receiver t_{PHL}

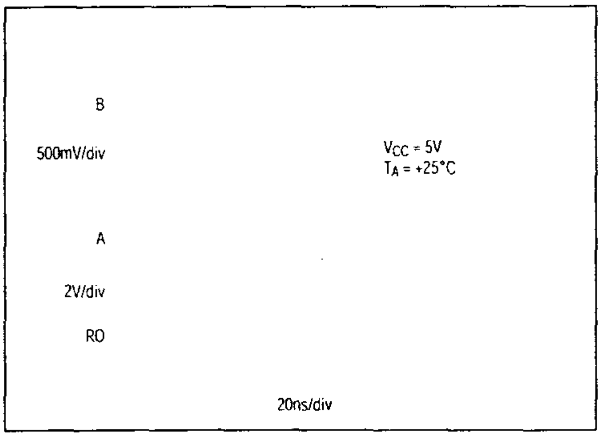


Figure 16. MAX481/MAX485/MAX490/MAX491/MAX1487 Receiver t_{PLH}

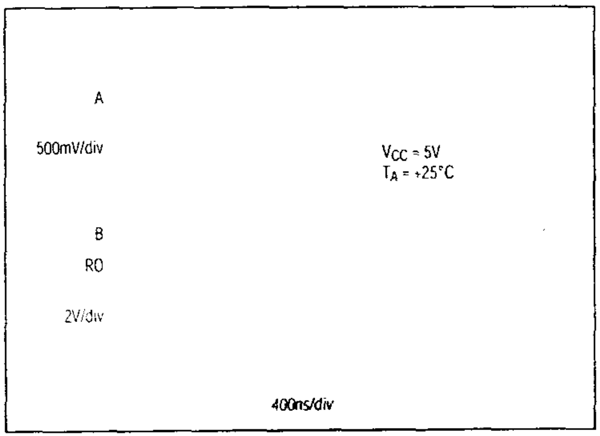


Figure 17. MAX483, MAX487–MAX489 Receiver t_{PHL}

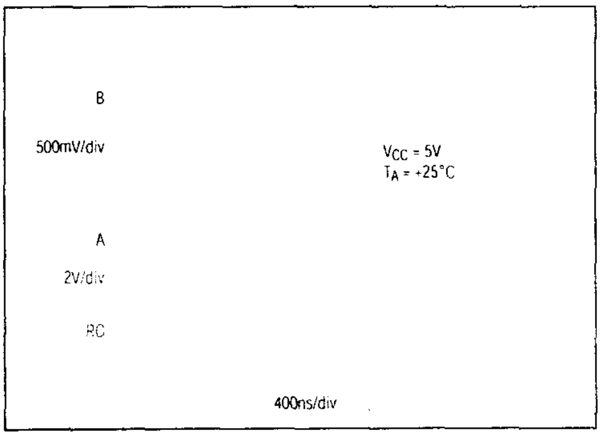


Figure 18. MAX483, MAX487–MAX489 Receiver t_{PLH}

Line Length vs. Data Rate

The RS-485/RS-422 standard covers line lengths up to 4000 feet. For line lengths greater than 4000 feet, see Figure 23.

Figures 19 and 20 show the system differential voltage for the parts driving 4000 feet of 26AWG twisted-pair wire at 110kHz into 120Ω loads.

Typical Applications

The MAX481, MAX483, MAX485, MAX487–MAX491, and MAX1487 transceivers are designed for bidirectional data communications on multipoint bus transmission lines. Figures 21 and 22 show typical network applications circuits. These parts can also be used as line repeaters, with cable lengths longer than 4000 feet, as shown in Figure 23.

To minimize reflections, the line should be terminated at both ends in its characteristic impedance, and stub lengths off the main line should be kept as short as possible. The slew-rate-limited MAX483 and MAX487–MAX489 are more tolerant of imperfect termination.

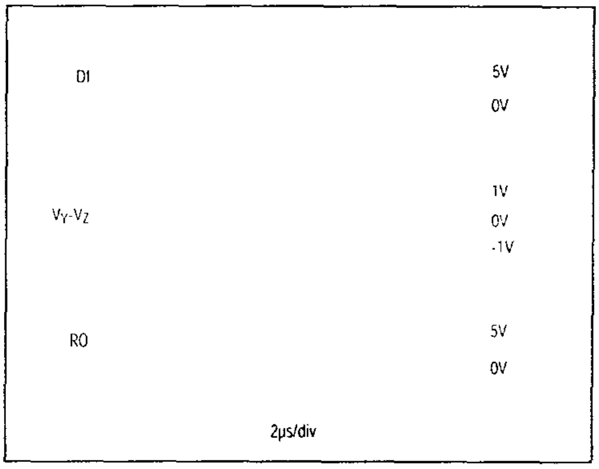


Figure 19. MAX481/MAX485/MAX490/MAX491/MAX1487 System Differential Voltage at 110kHz Driving 4000ft of Cable

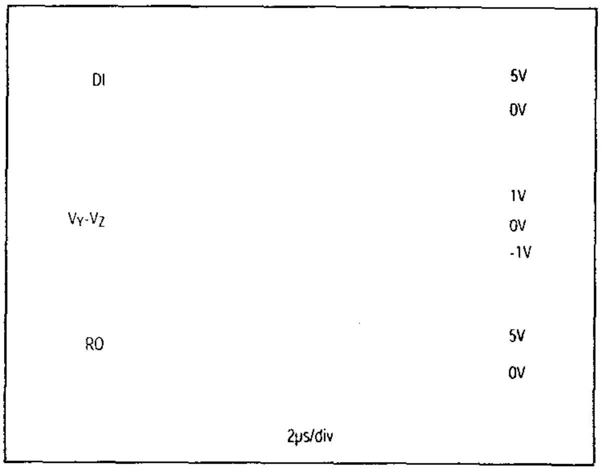


Figure 20. MAX483, MAX487–MAX489 System Differential Voltage at 110kHz Driving 4000ft of Cable

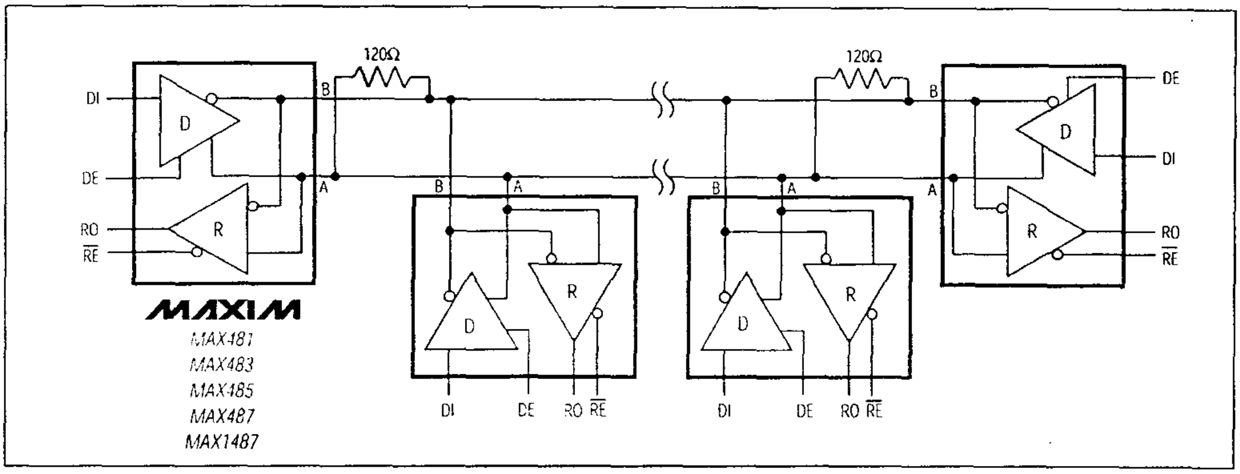


Figure 21. MAX481/MAX483/MAX485/MAX487/MAX1487 Typical Half-Duplex RS-485 Network

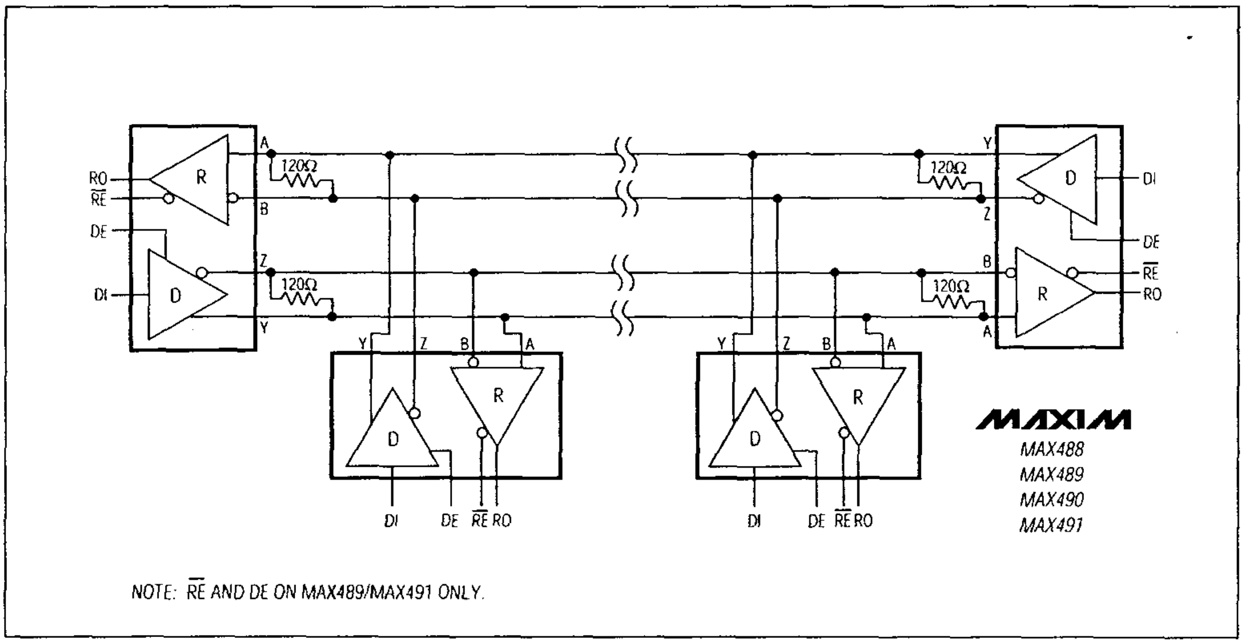


Figure 22. MAX488–MAX491 Full-Duplex RS-485 Network

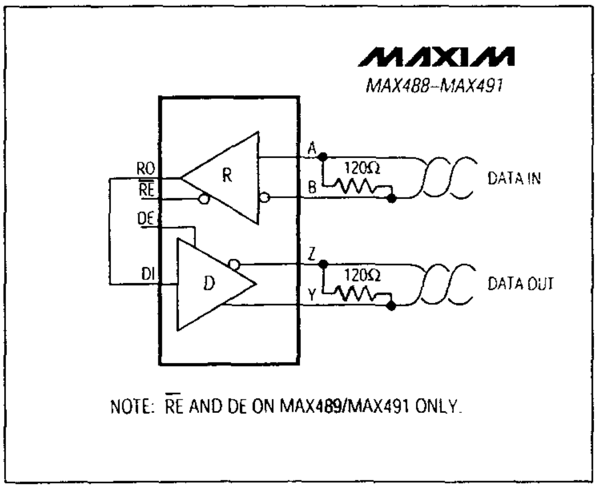


Figure 23. Line Repeater for MAX488–MAX491

Isolated RS-485

For isolated RS-485 applications, see the MAX253 and MAX1480 data sheets.

Ordering Information (continued)

PART	TEMP. RANGE	PIN-PACKAGE
MAX481EPA	-40°C to +85°C	8 Plastic DIP
MAX481ESA	-40°C to +85°C	8 SO
MAX481MJA	-55°C to +125°C	8 CERDIP
MAX483CPA	0°C to +70°C	8 Plastic DIP
MAX483CSA	0°C to +70°C	8 SO
MAX483CUA	0°C to +70°C	8 µMAX
MAX483C/D	0°C to +70°C	Dice*
MAX483EPA	-40°C to +85°C	8 Plastic DIP
MAX483ESA	-40°C to +85°C	8 SO
MAX483MJA	-55°C to +125°C	8 CERDIP
MAX485CPA	0°C to +70°C	8 Plastic DIP
MAX485CSA	0°C to +70°C	8 SO
MAX485CUA	0°C to +70°C	8 µMAX
MAX485C/D	0°C to +70°C	Dice*
MAX485EPA	-40°C to +85°C	8 Plastic DIP
MAX485ESA	-40°C to +85°C	8 SO
MAX485MJA	-55°C to +125°C	8 CERDIP
MAX487CPA	0°C to +70°C	8 Plastic DIP
MAX487CSA	0°C to +70°C	8 SO
MAX487CUA	0°C to +70°C	8 µMAX
MAX487C/D	0°C to +70°C	Dice*
MAX487EPA	-40°C to +85°C	8 Plastic DIP
MAX487ESA	-40°C to +85°C	8 SO
MAX487MJA	-55°C to +125°C	8 CERDIP
MAX488CPA	0°C to +70°C	8 Plastic DIP
MAX488CSA	0°C to +70°C	8 SO
MAX488CUA	0°C to +70°C	8 µMAX
MAX488C/D	0°C to +70°C	Dice*
MAX488EPA	-40°C to +85°C	8 Plastic DIP
MAX488ESA	-40°C to +85°C	8 SO
MAX488MJA	-55°C to +125°C	8 CERDIP
MAX489CPD	0°C to +70°C	14 Plastic DIP
MAX489CSD	0°C to +70°C	14 SO
MAX489C/D	0°C to +70°C	Dice*
MAX489EPD	-40°C to +85°C	14 Plastic DIP
MAX489ESD	-40°C to +85°C	14 SO
MAX489MJD	-55°C to +125°C	14 CERDIP

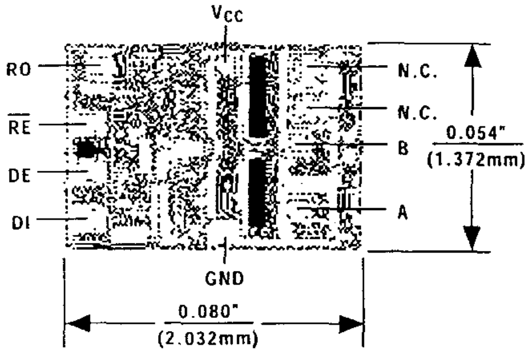
Ordering Information (continued)

PART	TEMP. RANGE	PIN-PACKAGE
MAX490CPA	0°C to +70°C	8 Plastic DIP
MAX490CSA	0°C to +70°C	8 SO
MAX490CUA	0°C to +70°C	8 µMAX
MAX490C/D	0°C to +70°C	Dice*
MAX490EPA	-40°C to +85°C	8 Plastic DIP
MAX490ESA	-40°C to +85°C	8 SO
MAX490MJA	-55°C to +125°C	8 CERDIP
MAX491CPD	0°C to +70°C	14 Plastic DIP
MAX491CSD	0°C to +70°C	14 SO
MAX491C/D	0°C to +70°C	Dice*
MAX491EPD	-40°C to +85°C	14 Plastic DIP
MAX491ESD	-40°C to +85°C	14 SO
MAX491MJD	-55°C to +125°C	14 CERDIP
MAX1487CPA	0°C to +70°C	8 Plastic DIP
MAX1487CSA	0°C to +70°C	8 SO
MAX1487CUA	0°C to +70°C	8 µMAX
MAX1487C/D	0°C to +70°C	Dice*
MAX1487EPA	-40°C to +85°C	8 Plastic DIP
MAX1487ESA	-40°C to +85°C	8 SO
MAX1487MJA	-55°C to +125°C	8 CERDIP

* *Contact factory for dice specifications.*

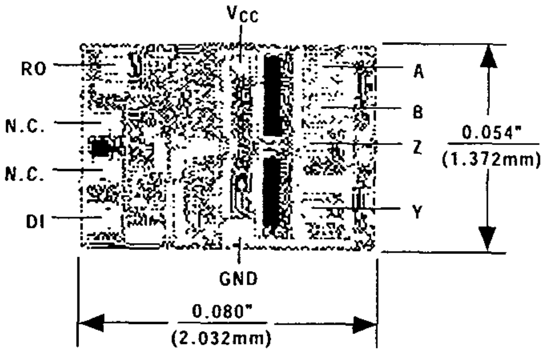
Chip Topographies

MAX481/MAX483/MAX485/MAX487/MAX1487

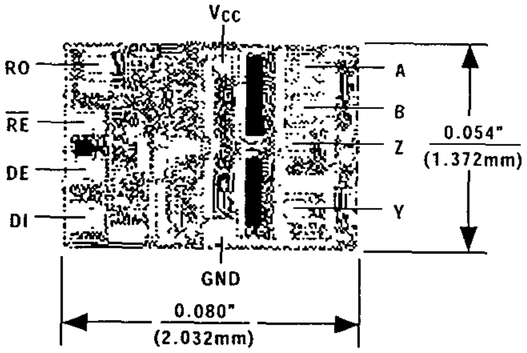


Chip Topographies (continued)

MAX488/MAX490

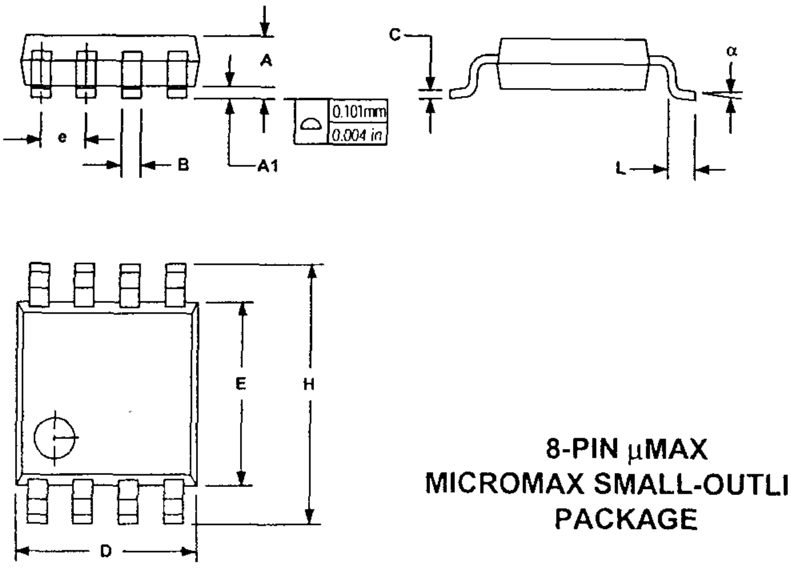


MAX489/MAX491



TRANSISTOR COUNT: 248
SUBSTRATE CONNECTED TO GND

Package Information



DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.036	0.044	0.91	1.11
A1	0.004	0.008	0.10	0.20
B	0.010	0.014	0.25	0.36
C	0.005	0.007	0.13	0.18
D	0.116	0.120	2.95	3.05
E	0.116	0.120	2.95	3.05
e	0.0256		0.65	
H	0.188	0.198	4.78	5.03
L	0.016	0.026	0.41	0.66
α	0°	6°	0°	6°

21-0036D

8-PIN µMAX
MICROMAX SMALL-OUTLINE
PACKAGE

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.