DAFTAR PUSTAKA

Almuchlisin, Agung Nugroho Jati, Umar Ali Ahmad. 2016. Perancangan dan Implementasi Sistem *Monitoring* Untuk Pelaporan Sampah Berbasis Teknologi *Embedded*. Yogyakarta : Teknik Elektro Sekolah Tinggi Teknolgi Adisutjipto. Vol.3, No.1 April 2016 | Page 666-673.

Andrianto, Heri. 2013. Pemrograman Microcontroller AVR ATMega16 Menggunakan Bahasa C. Bandung : Penerbit Informatika.

Data sheet. 2020. Diakses https://www.ti.com/lit/ds/symlink/l293.pdf tanggal 19 April 2020

Loadcell and Weight. (AmericaModule H:2010). https://www.ricelake.com diakses pada tanggal 10 April 2020.

Mardhiyah, Ajeng Wisnu. 2017. Rancang Bangun Alat Pemantau Gerak Manusia Menggunakan Kamera Berbasis *Personal Computer* (*Pc*). Surabaya : D3 Otomasi Sistem Instrumentasi Fakultas Vokasi Universitas Airlangga.

Moerdiyanto., 2009, Teknik monitoring dan evaluasi (monev) dalam rangka memperoleh informasi untuk pengambilan keputusan manajemen. Yogyakarta.

Muryanto, Joko (2009). Panduan Menggambar Schematic dan Mendesain PCB menggunakan Proteus 6 Profesional. Yogyakarta. Diakses dari http://jurnal.um-tapsel.ac.id/index.php/al muaddib/article/download/267/238 pada tanggal 15 April 2020.

R.Wrihatnolo, 2008, Monitoring, evaluasi, dan pengendalian : Konsep dan pembahasan. Yogyakarta.

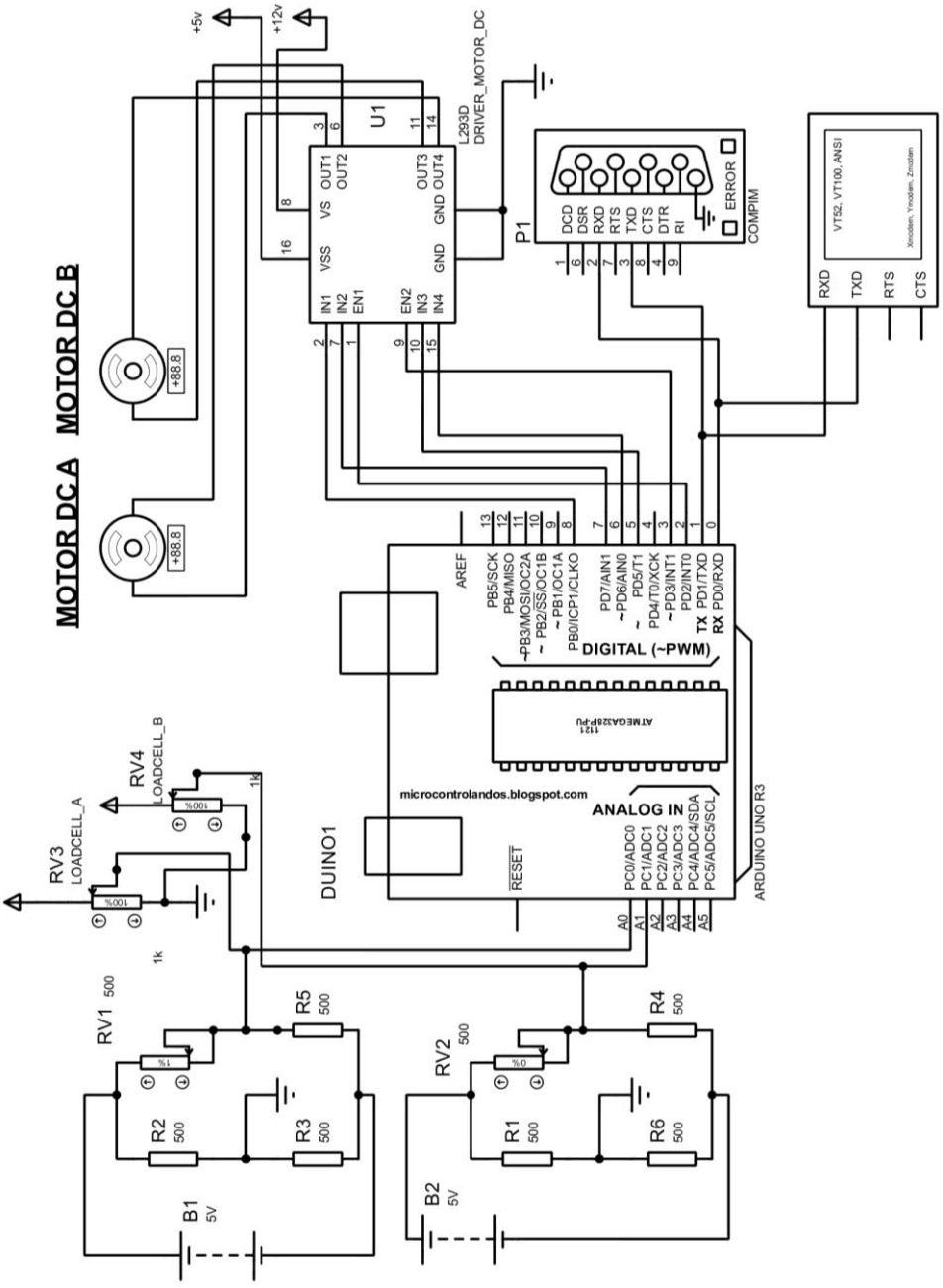
Rebby Fudi Alexander.2013."Aplikasi Sensor Berat Load Cell Pada Alat Pengering Herbal".

Renreng, Ilyas, 2012, "Rancang bangun dongkrak elektrik kapasitas 1 ton". Mekanika.Volume 3, No. 1, http://repository.unhas.ac.id/bitstream/handle/123456789/2493/JURNAL % 20ILYAS%20RENRENG.pdf?sequence=1, 10 April 2020.

Zaeni, Akhmad Mustofa. 2015. PROTEUS PROFESIONAL 8 SIMULASI & RANGKAIAN ELEKTRONIKA DASAR. Yogyakarta : Teknik Elektro Sekolah Tinggi Teknolgi Adisutjipto.

LAMPIRAN 1

Gambar rangkaian simulasi proteus



LAMPIRAN 2

Program Arduino IDE

```
#define Load Cella A0 //Pin ADC 1
#define Load Cellb A1 //pin ADC 2
#define motorA1 7 //pin output driver motor
#define motorA2 8 //pin output driver motor
#define motorB1 6 //pin output driver motor
#define motorB2 5 //pin output driver motor
#define enablea 3 //pin enable EN1 driver motor
#define enableb 2 //pin enable EN2 driver motor
int nilaiADC,nilaiADC2;
float volt,volt2;
//float kg,kg2;
void setup()
{
 Serial.begin(9600);
 pinMode(motorA1, OUTPUT);   // a1 merupakan pin, a2 dan b1 atau b2 juga
 pinMode(motorA2, OUTPUT);
 pinMode(motorB1, OUTPUT);
 pinMode(motorB2, OUTPUT);
 pinMode(enablea, OUTPUT);
 pinMode(enableb, OUTPUT);
 digitalWrite(enablea, HIGH);
 digitalWrite(enableb, HIGH);
}



void datadelphi()
```

```
{
  char cmd;
  cmd = Serial.read();
if(cmd=='w')
{
       digitalWrite(enablea, LOW);
       digitalWrite(enableb, LOW);
}
else if(cmd=='c')
{
       digitalWrite(enablea, HIGH);
       digitalWrite(enableb,HIGH);
}}
void Load Cell_a()
{
    if (nilaiADC==0)
    {
      digitalWrite(motorA1, HIGH);
       digitalWrite(motorA2, HIGH);
    }
    if (nilaiADC==1023)
    {
      digitalWrite(motorA1, LOW); // menentukan arah putaran jarum jam
       digitalWrite(motorA2, HIGH);
    }}
void Load Cell_b()
{
 if (nilaiADC2==0)
```

```
    {
      digitalWrite(motorB1, HIGH);
       digitalWrite(motorB2, HIGH);
    }
    if (nilaiADC2==1023)
    {
      digitalWrite(motorB1, LOW);
       digitalWrite(motorB2, HIGH);
    }  }
void loop()
{
 float nilaiADC = analogRead(Load Cella);
 float nilaiADC2 = analogRead(Load Cellb);

 float volt = nilaiADC*5/1023;
 float volt2 = nilaiADC2*5/1023;

float kg = volt;
float kg2 = volt2; // perhitungan dari voltase ke kg dengan perbandingan ( nilai voltase * ( 5 kg/ 5 volt), 0-5v ,0-1023, 0-5kg
Serial.print("a = "), Serial.println(kg), Serial.println("kg");
delay(1000);
Serial.print("b = "), Serial.println(kg2), Serial.println("kg");
delay(1000);
Load Cell_a();
Load Cell_b();
datadelphi();
```

LAMPIRAN 3

PROGRAM DELPHI XE 8 :

```
unit TA_1;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, CPort, strutils, Vcl.ExtCtrls;

type
  TForm2 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Button4: TButton;
    ComPort1: TComPort;
    Edit2: TEdit;
    Memo1: TMemo;
    Button5: TButton;
    Edit1: TEdit;
    Timer1: TTimer;
    Label2: TLabel;
    Edit3: TEdit;
    Edit4: TEdit;
    Memo2: TMemo;
    Label3: TLabel;
```

```
    Memo3: TMemo;
    Button3: TButton;
    Button6: TButton;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    Button7: TButton;
    procedure FormCanResize(Sender: TObject; var NewWidth, NewHeight:
Integer;
      var Resize: Boolean);
    procedure Button4Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormShow(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure Edit3Change(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure ComPort1RxChar(Sender: TObject; Count: Integer);
```

```
    procedure Memo3Change(Sender: TObject);
    procedure Memo1Change(Sender: TObject);
    procedure Memo2Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2; adc,berat : string;

implementation

{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);
begin
timer1.Enabled:=true;
comport1.Open;
ComPort1.WriteStr('c');//isi command adalah perintah ke arduino untuk memulai
motor
edit2.text:= 'MONITORING AKTIF!';
edit4.text:= 'MONITORING AKTIF!';
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
comport1.ShowSetupDialog;
end;
```

```
procedure TForm2.Button3Click(Sender: TObject);
begin
Memo2.Lines.SaveToFile('LOG_Sensor_A.txt');
end;

procedure TForm2.Button4Click(Sender: TObject);
begin
  if (application.messagebox('Aplikasi akan keluar secara paksa, pastikan apakah
PORT serial telah
ditutup?','PERINGATAN!!!1!!!111!!!11!!!1!',mb_yesno)=idyes)  then
   close;
end;

procedure TForm2.Button5Click(Sender: TObject);
begin
 case MessageDlg('Sistem akan dihentikan total, serial port dan motor akan tidak
aktif. Konfirmasi ?' ,
 mtConfirmation, [mbYes, mbNo], 0) of
  mrYes:
begin
  comPort1.WriteStr('w');//isi command adalah perintah ke arduino untuk reset
semua kondisi motor
  comPort1.Close;
  edit2.Color:=cLBlack;
  edit4.Color:=cLBlack;
  edit2.text:= 'SYSTEM SHUT DOWN!';
  edit4.text:= 'SYSTEM SHUT DOWN!';
end;
   mrNo: Abort;
end;
```

```
end;

procedure TForm2.Button6Click(Sender: TObject);
begin
Memo1.Lines.SaveToFile('LOG_Sensor_B.txt');
end;

procedure TForm2.Button7Click(Sender: TObject);
begin
comport1.Close;
edit2.text:= 'SERIALCOM CLOSED!';
edit4.text:= 'SERIALCOM CLOSED!';
end;

procedure TForm2.ComPort1RxChar(Sender: TObject; Count: Integer);
begin
begin
comport1.ReadStr(adc, count);
  begin
  berat:=berat+adc;
  memo3.text:=berat;
  end;
end;
end;

procedure TForm2.Edit1Change(Sender: TObject);
var
berat:integer;
begin
```

```
    berat:=strtoint(edit1.text);
    if berat=0 then
      begin
      edit2.Color:=clBlue;
      edit2.text:= 'STEADY STATE';
      end;
    if berat=1023 then
      begin
      edit2.Color:=clRed;
      edit2.text:= 'DISPOSING WASTE';
      end;
end;

procedure TForm2.Edit3Change(Sender: TObject);
var
berat:integer;
begin
    berat:=strtoint(edit3.text);
    if berat=0 then
      begin
      edit4.Color:=clBlue;
      edit4.text:= 'STEADY STATE';
      end;
    if berat=1023 then
      begin
      edit4.Color:=clRed;
      edit4.text:= 'DISPOSING WASTE';
      end;
end;
```

```
procedure TForm2.FormCanResize(Sender: TObject; var NewWidth,
  NewHeight: Integer; var Resize: Boolean);
begin
resize:=false;
end;

procedure TForm2.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
ReleaseCapture;
SendMessage(Form2.Handle,wm_SysCommand,$f012,0);
end;

procedure TForm2.FormShow(Sender: TObject);
begin
SetWindowPos(Self.Handle, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE or
SWP_NOSIZE or SWP_SHOWWINDOW);
end;

procedure TForm2.Memo1Change(Sender: TObject);
begin
memo1.perform( WM_VSCROLL, SB_BOTTOM, 0 );
memo1.perform( WM_VSCROLL, SB_ENDSCROLL, 0 );
end;

procedure TForm2.Memo2Change(Sender: TObject);
begin
memo2.perform( WM_VSCROLL, SB_BOTTOM, 0 );
memo2.perform( WM_VSCROLL, SB_ENDSCROLL, 0 );
```

```
end;

procedure TForm2.Memo3Change(Sender: TObject);
var rin:string;
begin
 rin:=memo3.Lines[memo3.Lines.count-2];
 if rightstr(rin,1)= 'a' then
    begin
    edit1.Text:=leftstr(rin,length(rin)-1);
    strtoint(edit1.text);
    memo1.Lines.Add(edit1.Text);
    end;
  if rightstr(rin,1)= 'b' then
    begin
    edit3.Text:=leftstr(rin,length(rin)-1);
    strtoint(edit3.text);
    memo2.Lines.Add(edit3.Text);
    end;
memo3.perform( WM_VSCROLL, SB_BOTTOM, 0 );
memo3.perform( WM_VSCROLL, SB_ENDSCROLL, 0 );
end;

end.
```

LAMPIRAN 4

Data Sheet Driver Motor L293D

TEXAS INSTRUMENTS

L293, L293D
SLRS008D – SEPTEMBER 1986 – REVISED JANUARY 2016

L293x Quadruple Half-H Drivers

1 Features

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- High-Noise-Immunity Inputs
- 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)

2 Applications

- Stepper Motor Drivers
- DC Motor Drivers
- Latching Relay Drivers

3 Description

The L293 and L293D devices 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.

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.

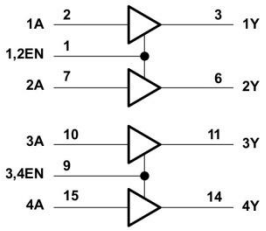
The L293 and L293D are characterized for operation from 0°C to 70°C.

Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
L293NE	PDIP (16)	19.80 mm × 6.35 mm
L293DNE	PDIP (16)	19.80 mm × 6.35 mm

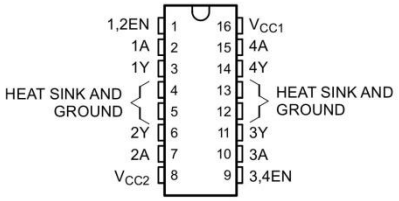
(1) For all available packages, see the orderable addendum at the end of the data sheet.

Logic Diagram



5 Pin Configuration and Functions

NE Package
16-Pin PDIP
Top View



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, noninverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to printed-circuit-board ground plane with multiple solid vias
V_{CC1}	16	—	5-V supply for internal logic translation
V_{CC2}	8	—	Power VCC for drivers 4.5 V to 36 V

6 Specifications

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)(1)

	MIN	MAX	UNIT
Supply voltage, V_{CC1}(2)		36	V
Output supply voltage, V_{CC2}		36	V
Input voltage, V_I		7	V
Output voltage, V_O	–3	V_{CC2} + 3	V
Peak output current, I_O (nonrepetitive, t ≤ 5 ms): L293	–2	2	A
Peak output current, I_O (nonrepetitive, t ≤ 100 µs): L293D	–1.2	1.2	A
Continuous output current, I_O: L293	–1	1	A
Continuous output current, I_O: L293D	–600	600	mA
Maximum junction temperature, T_J		150	°C
Storage temperature, T_{stg}	–65	150	°C

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) All voltage values are with respect to the network ground terminal.

6.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001(1)	±2000	V
		Charged-device model (CDM), per JEDEC specification JESD22-C101(2)	±1000	

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

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

			MIN	NOM	MAX	UNIT
	Supply voltage	V_{CC1}	4.5		7	V
		V_{CC2}	V_{CC1}		36	
V_{IH}	High-level input voltage	V_{CC1} ≤ 7 V	2.3		V_{CC1}	V
		V_{CC1} ≥ 7 V	2.3		7	V
V_{IL}	Low-level output voltage		–0.3(1)		1.5	V
T_A	Operating free-air temperature		0		70	°C

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

6.4 Thermal Information

THERMAL METRIC(1)		L293, L293D NE (PDIP) 16 PINS	UNIT
$R_{\theta JA}$	Junction-to-ambient thermal resistance (2)	36.4	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	22.5	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	16.5	°C/W
ψ_{JT}	Junction-to-top characterization parameter	7.1	°C/W
ψ_{JB}	Junction-to-board characterization parameter	16.3	°C/W

(1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, SPRA953.
(2) The package thermal impedance is calculated in accordance with JESD 51-7.

6.5 Electrical Characteristics

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

PARAMETER			TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage		L293: I_{OH} = −1 A		V_{CC2} – 1.8	V_{CC2} – 1.4		V
			L293D: I_{OH} = − 0.6 A					
V_{OL}	Low-level output voltage		L293: I_{OL} = 1 A			1.2	1.8	V
			L293D: I_{OL} = 0.6 A					
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	

6.6 Switching Characteristics

over operating free-air temperature range (unless otherwise noted) V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER			TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output from A input	L293NE, L293DNE	C_L = 30 pF, See Figure 2		800		ns
		L293DWP, L293N L293DN			750		
t_{PHL}	Propagation delay time, high-to-low-level output from A input	L293NE, L293DNE			400		ns
		L293DWP, L293N L293DN			200		
t_{TLH}	Transition time, low-to-high-level output	L293NE, L293DNE			300		ns
		L293DWP, L293N L293DN			100		
t_{THL}	Transition time, high-to-low-level output	L293NE, L293DNE			300		ns
		L293DWP, L293N L293DN			350		

6.7 Typical Characteristics

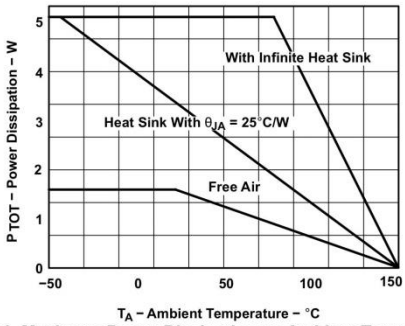
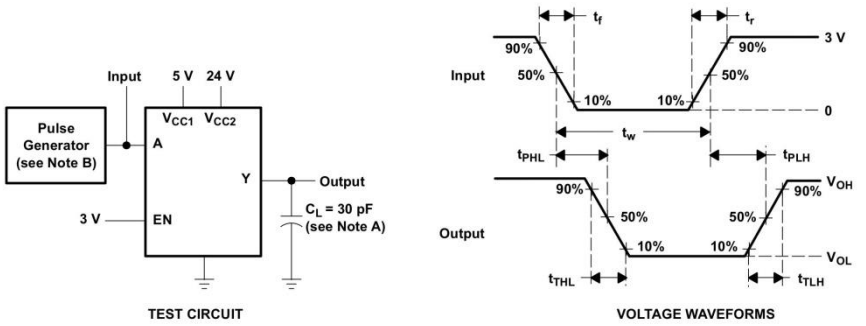


Figure 1. Maximum Power Dissipation vs Ambient Temperature

7 Parameter Measurement Information



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, $Z_O = 50$ Ω.

Figure 2. Test Circuit and Voltage Waveforms