

PERCOBAAN 2 :
PENGENDALIAN MOTOR DC
MENGUNAKAN INTERFACE PPI 8255

PENGENDALIAN MOTOR DC MENGUNAKAN INTERFACE PPI 8255

TUJUAN :

1. Mampu memahami dan membedakan pengendalian motor DC, yaitu single pulse, High speed pulse sequence dan PWM (Pulse Width Modulation).
2. Mengetahui cara pengukuran kecepatan motor (Rotation Speed) melalui metode digital dengan kendali single pulse.
3. Mampu membuat program sederhana berbasis windows (GUI) untuk mengendalikan motor DC melalui PC.

ALAT-ALAT :

1. 1 unit PC compatible.
2. 1 buah Oscilloscope.
3. 1 buah Interface Card ISA to PPI 8255 (IB-1009).
4. 1 buah Modul dc Motor Control (U-1009C).
5. 1 buah Power Supply eksternal/Bread Board (U-1009D).

MATERI:

1. Teknik mempercepat dan memperlambat putaran Motor DC.
2. Pengukuran kecepatan putaran Motor DC.

KONSEP DASAR :

Motor DC memiliki beberapa kekurangan dibandingkan dengan Motor Stepper yang derajat perputaran dan kecepatannya dapat dikendalikan dengan baik. Namun Motor DC juga memiliki kelebihan lain yaitu torsi yang lebih besar ketimbang Motor Stepper. Untuk mengendalikan kecepatan Motor DC, digunakan suatu metode PWM (Pulse Width Modulation) yaitu teknik pemberian daya berupa pulsa ke Motor DC. Selain itu pula PWM Driver ini dapat menentukan arah pergerakan Motor DC, apakah searah jarum jam (ClockWise) atau berlawanan arah jarum jam (CounterClockWise). Kecepatan Motor DC ini juga dapat diperkirakan dengan menggunakan paduan RPM Encoder Disk, Photo Detector dan Counter.

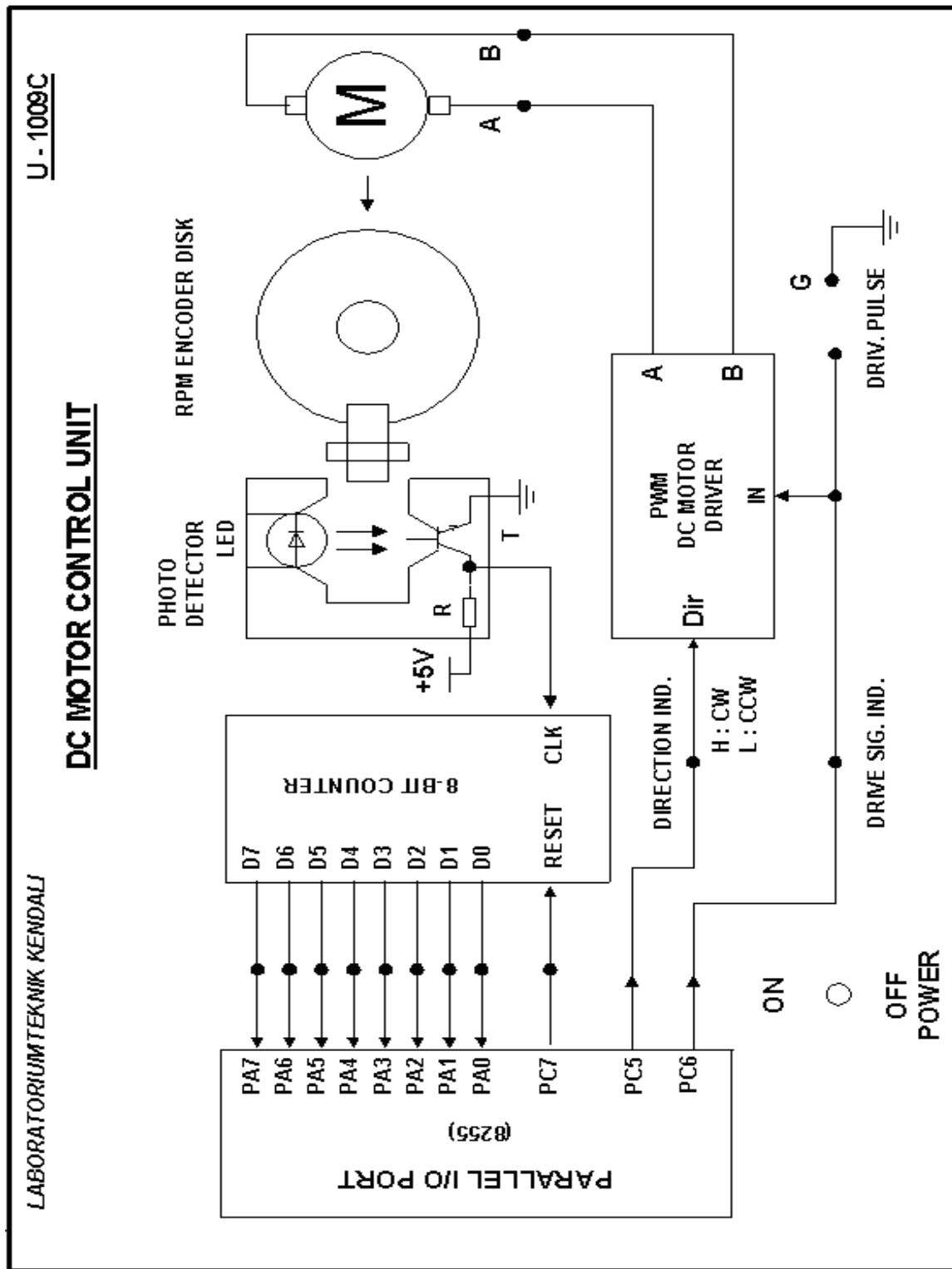
PEMAHAMAN MODUL U-1009C :

Pengendalian Motor DC :

1. POWER SW. : saklar daya.
2. ON LED : jika saklar daya aktif LED akan menyala.
3. DIRECTION IND. : Ketika motor DC berotasi searah jarum jam (CW), maka LED akan menyala, ketika berotasi berlawanan arah LED tidak menyala.
4. DRIV SIG. IND. : LED yang menunjukkan kondisi pulsa untuk pengendalian motor DC, kecerahan cahaya LED sesuai pada siklus tugas selama pengendalian PWM.
5. DRIVE PULSE : Titik pengujian untuk mengamati gelombang pulsa (Pulse Wave) untuk pengendalian motor DC.

6. LED (PA0-PA7) : Menunjukkan nilai keluaran dari counter 8 bit, digunakan untuk mengukur kecepatan rotasi motor.
7. LED (PC7) : Ketika LED berubah dari mati menjadi menyala counter diinisialisasi ke 0 dan mulai menghitung jumlah pulsa yang diterapkan pada CLK. Ketika LED mati, counter menahan nilainya.
8. PHOTO DETECTOR : Ketika motor berotasi 20° , satu pulsa dihasilkan, maka 18 pulsa akan dihasilkan untuk satu rotasi.

Gambar rangkaian Modul U-1009C



PROSEDURE PERCOBAAN & DATA PENGAMATAN :

1. TEKNIK MEMPERCEPAT & MEMPERLAMBAT PUTARAN MOTOR DC.

- a. Hidupkan PC, Oscilloscope dan Modul.
- b. Hubungkan CH1 Oscilloscope pada terminal DRIV. PULSE.
- c. Klik program MDCM01.PAS dan jalankan program.
- d. Tes program dengan memasukkan nilai 100 pada rasio PWM. (*cat : bila motor tidak berputar lewatkan dulu ke percobaan 2*)
- e. Atur settingan hingga terdapat tampilan pada oscilloscope.
- f. Tegangan puncak yang terukur pada oscilloscope : (....) volt
- g. Setting Oscilloscope yang dipakai : volt = (....) volt/div
Time = (....) time/div
- h. Isikan PWM (ON/T) sesuai dengan data pada tabel MDC-1.
- i. Amati gambar bentuk gelombang yang ditampilkan pada oscilloscope (t₀ – t₈ merupakan rentang untuk 2 siklus gelombang penuh).
- j. Perhatikan kecepatan rotasi motor yang terjadi untuk setiap siklus.

Rasio PWM (ON/T)	<i>Bentuk signal dari oscilloscope</i>
0 (0%)	
10 (25%)	
20 (50%)	
30 (75%)	
40 (100%)	

Table MDC-1

Isi Table diatas dengan seksama.

Pertanyaan :

Bagaimana kecepatan putaran motor DC, apabila terjadi perubahan Rasio PWM dari 0% menjadi 75% ?

Bagaimana kecepatan putarannya motor DC, apabila terjadi perubahan Rasio PWM dari 100% menjadi 25% ?

Dari percobaan yang anda lakukan, bagaimana caranya untuk mempercepat dan memperlambat putaran pada motor DC ? Jelaskan !

2. PENGUKURAN KECEPATAN PUTARAN MOTOR DC.

- Buka dan jalankan program DCM03.PAS.
- Isikan waktu tunda dengan nilai terserah anda (1–10000), sebelum tombol start ditekan perhatikan modul DCM dan perhatikan kondisi LED PC7.

Pengamatan : (*coret yang tidak perlu*)

Kondisi LED PC7 pada saat motor mulai bergerak : (High/Low).

Kondisi LED PC7 pada saat motor akan berhenti : (High/Low)

- Lakukan pengamatan untuk 5 data yang terdapat pada table MDC-2

Ketentuan :

- ❖ Pulsa dan RPM dilihat dari layar monitor.
- ❖ Catat data D7.....D0 pada table pengamatan (*D7-D0 = dilihat dari modul DCM, 0=mati ; 1=hidup*)
- ❖ Desimal didapat dari D7.....D0 yang telah dikonversi kedalam bilangan Biner ($D7 * 2^7 + D6 * 2^6 + + D0 * 2^0$)

No	Waktu Tunda	Pulsa	RPM	D7	D6	D5	D4	D3	D2	D1	D0	Desimal
1	1											
2	10											
3	100											
4	1000											
5	10000											

Table MDC-2

Dengan menggunakan rumus $RPM = (D/18[\text{lubang pada encoder disk}]) \times (1/0,2) \times 60[1 \text{ menit} = 60 \text{ detik}]$ dan D = jumlah pulsa (harga D diambil dari nilai desimalnya), tentukan harga RPM dari data ke-3 (waktu tunda 100) ! *cat : kecepatan maximum modul DCM = 4250rpm*

Paraf Asisten dengan Nama Jelas :

1	2	M
---	---	---

Syarat syahnya laporan ini !!!

MODIFIKASI PROGRAM DELPHI :

1. Tampilan pada form yang diinginkan :

The screenshot shows a Delphi form titled "Form 1". Inside the form, there is a group box labeled "arah" containing two radio buttons: "CW" (selected) and "CCW". To the right of the group box is a label "Rasio PWM (ON/T)" followed by an empty text box. At the bottom of the form, there are three buttons: "START", "STOP", and "EXIT".

2. Penjelasan untuk output yang diinginkan :

This is a large, empty rectangular box provided for the user to write a detailed explanation of the expected output of the program.

3. Setting properties tools yang dibutuhkan :

KOMPONEN	PROPERTIES	INPUT VALUE
Button1	Caption	Start
Button2	Caption	Exit
Button3	Caption	Stop
RadioGroup1	Caption Items.line 1 Items.line 2 Name	Arah CW CCW rg
Label1	Caption	Rasio PWM (ON/T)
Edit1	Text	
Timer1	Enabled Interval	False 100
SmallPort1	Opened Name	True SP

4. Code Editor :

```

button1                                // Start //
begin
timer1.enabled := .....;
end;

button3                                // Exit //
begin
.....;
end;

button2                                // Stop //
begin
timer1.enabled := .....;
end;

timer
begin
if ..... (edit1.text) > ..... then
// jika PWM lebih dari 100 maka program diam //
begin
timer1 ..... := false;
end
else
begin
sp.port [.....] := .....;           // Inisialisasi //
if rg.itemindex = 0 then             // Jika arahnya cw maka //
begin
sp.port [.....] := .....;           // arah cw //
sp.port [.....] := .....;           // onc //
sp.delay(strtoint(.....));           // PWM //
sp.port [.....] := .....;           // ofc //
sp.delay (.....-(.....(.....)));     // delay onc + delay ofc = 100 //
end
else if rg.itemindex = 1 then         // Jika arahnya ccw maka //
begin
sp.port [.....] := .....;           // arah ccw //
sp.port [.....] := .....;           // onc //
sp.delay(strtoint(.....));           // PWM //
sp.port [.....] := .....;           // ofc //
sp.delay (.....-(.....(.....)));     // delay onc + delay ofc = 100 //
end
else
begin
timer1 ..... := .....;             // jika tidak memilih arah maka motor diam //
end;
end;
end;

```


JAWABAN LAPORAN PENDAHULUAN :

This image shows a full page of primary-ruled paper. It features a solid vertical line on the left side, creating a margin. The rest of the page is filled with horizontal dotted lines, providing a guide for handwriting practice. There are no other markings or text on the page.

This image shows a full page of primary-ruled paper. It features a solid vertical line on the left side, creating a margin. The rest of the page is filled with horizontal dotted lines, providing a guide for handwriting practice. There are no other markings or text on the page.

SOAL LAPORAN AKHIR :

[illegible]

JAWABAN LAPORAN AKHIR :

[illegible]

