

# PERCOBAAN 11

## PULSE WIDTH MODULATION

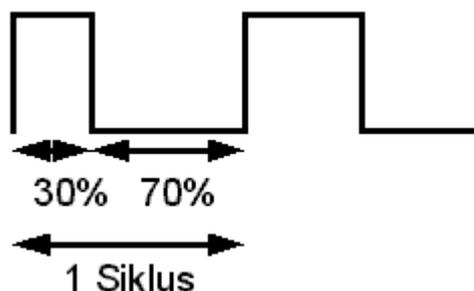
### TUJUAN:

1. Memahami prinsip dasar PWM
2. Memahami rangkaian Driver Motor DC
3. Memahami pemrograman assembly untuk pengaturan PWM

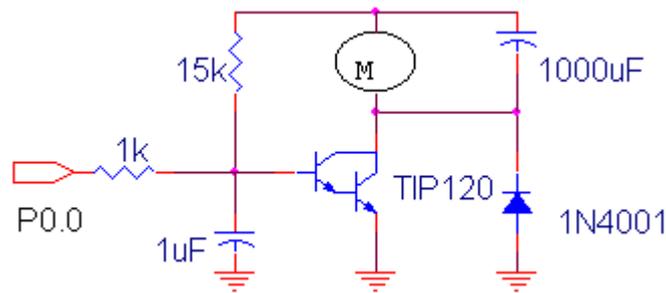
### Konsep Dasar PWM

Salah satu cara yang paling mudah untuk membangkitkan sebuah tegangan analog dari sebuah nilai digital adalah dengan menggunakan pulse-width modulation (PWM). Dalam PWM gelombang kotak, frekuensi tinggi dibangkitkan sebagai output digital. Untuk contoh, sebuah port bit secara kontinu melakukan kegiatan saklar on dan off pada frekuensi yang relatif tinggi. Selanjutnya, bila sinyal diumpankan pada LPF low pass filter, tegangan pada output filter akan sama dengan Root Mean Square ( RMS ) dari sinyal gelombang kotak. Selanjutnya tegangan RMS dapat divariasikan dengan mengubah duty cycle dari sinyal.

DUTY CYCLE menyatakan fraksi waktu sinyal pada keadaan logika high dalam satu siklus. Satu siklus diawali oleh transisi low to high dari sinyal dan berakhir pada transisi berikutnya. Selama satu siklus, jika waktu sinyal pada keadaan high sama dengan low maka dikatakan sinyal mempunyai DUTY CYCLE 50 %. DUTY CYCLE 20 % menyatakan sinyal berada pada logika 1 selama 1/5 dari waktu total



Gambar 11.1. Duty cycle 30 %



Gambar 11.2. Aplikasi PWM pada setting kecepatan motor

### PWM dengan Mikrokontroler

Pada rangkaian tersebut menunjukkan sebuah DAC yang dibangun dengan metode PWM, yang digunakan untuk mengendalikan kecepatan motor DC dengan modulasi lebar pulsa. Bit 0 dari P0 mengemudikan sebut saklar transistor sebagaimana yang ditunjukkan pada gambar. Motor dihidupkan dan dimatikan untuk suatu periode tertentu

Bagian pada saat motor hidup disebut DUTY CYCLE. Pada program ini menggunakan sebuah byte untuk menyimpan lama waktu motor on, dari sejumlah 256 siklus. Bila duty cycle yang diberikan adalah 10 % maka program ini menyimpan data waktu ON selama 10 siklus dan OFF selama 246 siklus dari 256 siklus.

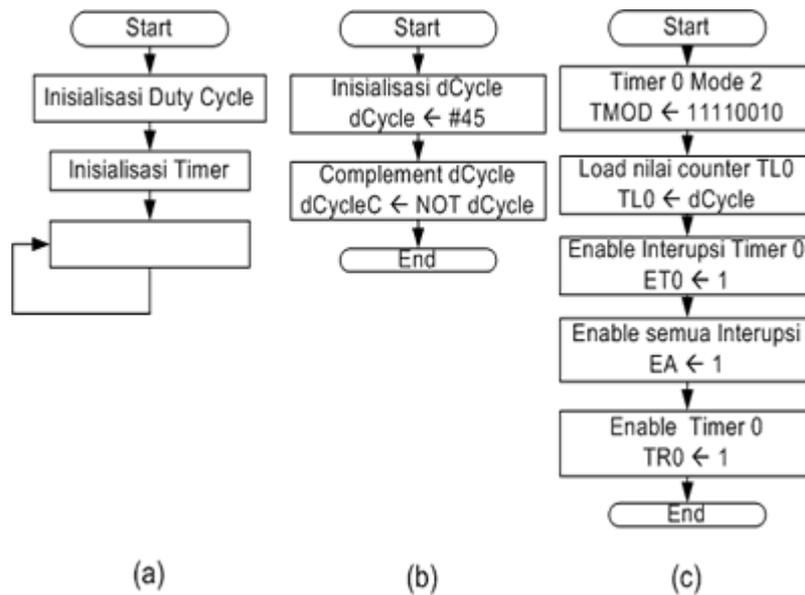
Berikutnya nilai duty cycle disimpan pada RAM internal yang diberi label dCycle. Komplemen dari duty cycle disimpan pada RAM internal dengan nama dCycleC. Pada perancangan software ini, Timer 0 diaplikasikan dalam mode 2, yaitu 8 bit timer auto reload, yang akan melakukan increment nilai register counter setiap siklus, dan bila terjadi overflow maka data yang berada pada TH0 akan diloadkan ke TL0 yang berfungsi sebagai counter 8 bit.

Bila frekuensi kristal yang digunakan adalah 12 Mhz, sehingga jika nilai reload adalah 0 maka timer 0 akan over flow setiap 256 udetik; dan jika nilai reload adalah FFh maka timer akan over flow setiap 1 udetik. Pertama kali program menghidupkan motor dan menempatkan nilai dCycle ke TH0 sebagai nilai reload. Setelah timer overflow, komplemen dari duty cycle dCycleC akan ditempatkan ke TH0 sebagai nilai reload dan motor berhenti berputar. Pada pemrograman ini keadaan motor dapat dilihat pada register yang dapat dialamati bit, yang ditandai sebagai motorFlag.

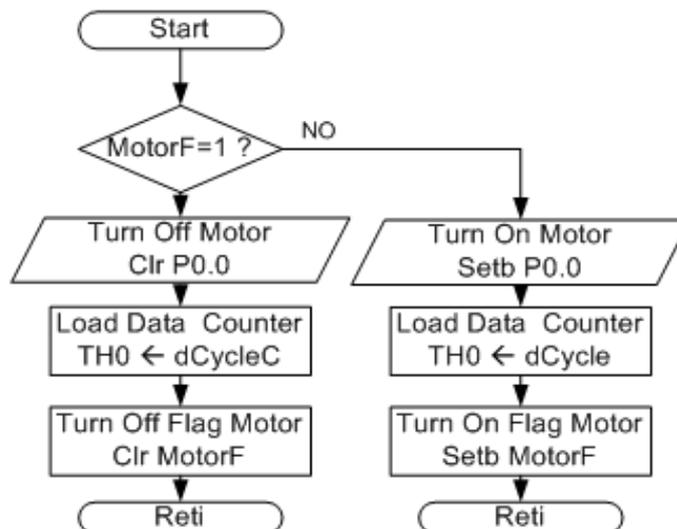
## Percobaan 11.1.

### Setting kecepatan putaran motor DC dengan PWM

Pada percobaan ini, putaran kecepatan motor kemudian dengan menggunakan transistor TIP120 melalui metode PWM. PWM kemudian dengan menggunakan satu bit dari P0, seperti yang ditunjukkan pada gambar dibawah.



Gambar 11.3. Diagram alir (a) rutin utama (b) inisialisasi duty cycle dan (c) inisialisasi timer pada percobaan 11.1



Gambar 11.4. Diagram alir interupsi timer 0 pada percobaan 11.1

1. Hubungkan kabel data dan ke inputan rangkaian driver motor DC
2. Hubungkan modul Microcontroller Trainer dengan power supply +5V
3. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
4. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
5. Ketik program berikut ini:

```

dCycle      equ 30h
dCycleC     equ 31h
PWM         bit P0.0
MotorF      bit 20h

    org 0h
    sjmp start
    org 0bh
    ljmp Interupsi_Timer0
;

start: call Init_Dcycle
      call init_Timer
loop:  sjmp loop
;

Init_Dcycle:
      mov dCycle,#45    ;inisialisasi data dCycle T_ON
      mov A,dCycle     ;copy data ke A
      cpl A           ;komplemen A
      mov dCycleC,A    ;copy A ke dCycleC
      ret
;

Init_Timer:
      anl TMOD,#0F0h ;
      orl TMOD,#2    ;Timer0 mode2 auto reload
      mov TH0,dCycle ;Load data counter 8 bit
      setb ET0      ;enable interupsi timer0
      setb EA       ;enable all interupsi
      setb TR0      ;timer0 run
      ret
;

Interupsi_Timer0:
      jb motorF,motorOff ;deteksi bit motorF

```

```

        setb PWM            ;Hidupkan motor
        setb P0.7          ;TP pulsa PWM di osiloskop
        mov TH0,dCycle     ;load data dCycle ke TH0
        setb motorF       ;beri tanda motorF=1-> motor ON
        reti

motorOff:
        clr PWM           ;matikan motor
        clr P0.7         ;TP pulsa PWM di osiloskop
        mov TH0,dCycleC   ;load data cCycleC ke TH0
        clr motorF       ;beri tanda motorF=0-> motor OFF
        reti
        end

```

6. Simpanlah program yang anda ketik dan beri nama : prog151a.asm
7. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari \*.asm ke \*.hex.
8. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software ( Lihat Petunjuk Penggunaan)

## Percobaan 11.2.

### Pengaturan kecepatan putaran motor DC dengan metode PWM melalui ADC

Pada percobaan ini, ADC digunakan untuk membaca tegangan dari input pembagi tegangan 0-5 Volt, selanjutnya tegangan yang telah diubah menjadi data digital, digunakan untuk mengatur PWM pada Motor DC.

1. Hubungkan kabel data dan ke inputan rangkaian driver motor DC
2. Hubungkan modul Microcontroller Trainer dengan power supply +5V
3. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
4. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
5. Ketik program berikut ini:

```
dCycle equ 30h
dCycleC equ 31h
PWM bit P0.0
MotorF bit 20h
    org 0h
    sjmp start
    org 0bh
    ljmp Interupsi_Timer0
;
start: call init_Timer
loop:  call ADC
      mov dCycle,A
      cpl A
      mov dCycleC,A
      sjmp loop
;
ADC:  clr P3.3
      nop
      nop
      nop
      setb P3.3
eoc:  jb P3.2,eoc
      clr P3.4
      mov A,P1
      setb P3.4
      ret
```

```

;
Init_Timer:
    mov dCycle,#40
    mov A,dCycle
    cpl A
    mov dCycleC,A
    anl TMOD,#0F0h
    orl TMOD,#2
    mov TH0,dCycle
    setb ET0
    setb EA
    setb TR0
    ret
;
Interupsi_Timer0:
    jb motorF,motorOff
    setb PWM
    setb P0.7
    mov TH0,dCycle
    setb motorF
    reti
motorOff:
    clr PWM
    clr P0.7
    mov TH0,dCycleC
    clr motorF
    reti
end

```

6. Simpanlah program yang anda ketik dan beri nama : prog151a.asm
7. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari \*.asm ke \*.hex.
8. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software ( Lihat Petunjuk Penggunaan)