JOBSHEET VIII
MENGGUNAKAN *TIMER/COUNTER* DALAM MIKROKONTROLER ATMEGA8535

TUJUAN

- Mahasiswa mampu menggunakan fitur timer/counter mikrokontroler.
- Mahasiswa mampu menggunakan mikrokontroler untuk membuat timer.
- Mahasiswa mampu menggunakan mikrokontroler untuk menghitung banyaknya pulsa yang masuk.

TIMER/COUNTER

Timer/counter dalam ATmega8535 ada 3 yaitu:

- Timer/counter 0
- Timer/counter 1
- Timer/counter 2

Interrupt timer berasal dari dua sumber yaitu:

- Overflow interrupt, dimana interrupt terjadi jika TCNTn mencapai 255 untuk timer 8 bit dan 65535 untuk timer 16 bit.
- Compare match interrupt, dimana interrupt terjadi jika nilai OCR sama dengan TCNTn.

Secara umum fitur Timer/Counter mikrokontroler ATmega8535 dapat digunakan untuk berbagai macam fungsi, yaitu:

- **Timer/delay time**

Pada dasarnya ketika Timer/Counter difungsikan sebagai Timer, sistem hanya menghitung pulsa clock. Frekuensi pulsa clock yang dihitung tersebut bisa sama dengan frekuensi kristal yang digunakan atau dapat diperlambat menggunakan prescaler dengan faktor 8, 64, 256, atau 1024. Contohnya jika sebuah sistem mikrokontroler menggunakan kristal dengan frekuensi 4 MHz dan timer yang digunakan adalah timer 8 bit, maka maksimum waktu timer yang bisa dihasilkan adalah:

$$t_{MAX} = \frac{1}{f_{clk}} x\ (FFh + 1)$$
$$t_{MAX} = \frac{1}{4000000}\ x\ (255 + 1)$$
$$t_{MAX} = 0{,}000064\ s$$

Untuk menghasilkan timer yang lebih lama dapat digunakan prescaler, misalnya 1024, maka maksimum waktu timer yang bisa dihasilkan adalah:

$$t_{MAX} = \frac{1}{f_{clk}} x\ (FFh + 1)\ x\ N$$
$$t_{MAX} = \frac{1}{4000000}\ x\ (255 + 1)\ x\ 1024$$
$$t_{MAX} = 0{,}065536\ s$$

Untuk menghitung nilai TCNT supaya menghasilkan waktu timer tertentu dipergunakan rumus berikut:

$$TCNT = (1 + FFh) - \frac{(T_{timer} \, x \, f_{clk})}{N}$$

Dimana:
TCNT = nilai Timer (Heksadesimal)
fCLK = Frekuensi clock kristal yang digunakan (Hz)
Ttimer = Waktu timer yang diinginkan (detik)
N = prescaler (1,8,64,256,1024)
1+FFh = nilai maksimum timer adalah FFh dan overflow saat FFh ke 00h

- **Counter**

Secara prinsip, memfungsikan Timer/Counter sebagai Counter sama dengan fungsi sebagai Timer akan tetapi sumber clock bukan berasal dari frekuensi kristal, tetapi input dari kaki Tn. Dengan memanfaatkan counter naik, maka bisa diberikan nilai TCNT yang sesuai dengan rumus berikut:

TCNT = (1+FFH) – jumlah counter

Sehingga jika diinginkan membuat counter 5, maka nilai TCNT adalah FBH.

- Mode CTC

Dengan mode ini, maka mikrokontroler bisa membangkitkan sinyal dengan frekuensi tertentu sesuai dengan rumus berikut:

$$f_{OCn} = \frac{f_{clk_I/O}}{2.N.(1 + OCRn)}$$

Jika diinginkan untuk membangkitkan sinyal dengan frekuensi 1 kHz, maka dengan frekuensi clock 4 MHz, dan N=16, maka diperoleh nilai OCR sebagai berikut:

$$f_{OCn} = \frac{f_{clk_I/O}}{2.N.(1 + OCRn)}$$

$$1 + OCRn = \frac{f_{clk_I/O}}{2.N.f_{OCn}}$$

$$OCRn = \frac{f_{clk_I/O}}{2.N.f_{OCn}} - 1$$

$$OCRn = \frac{4000000}{2.16.1000} - 1$$

$$OCRn = 124$$

- Mode PWM

Dengan mode ini, maka mikrokontroler bisa membangkitkan sinyal dengan frekuensi tertentu dan duty cycle tertentu sesuai dengan rumus sebagai berikut.

Untuk Fast PWM:

$$f_{OCnPWM} = \frac{f_{clk_I/O}}{N.256}$$

Untuk Phase Correct PWM:

$$f_{OCnPWM} = \frac{f_{clk_I/O}}{2.N.256}$$

Dimana nilai OCR yang berkisar dari 0 sampai 255 akan mempengaruhi duty cycle sinyal yang dibangkitkan.

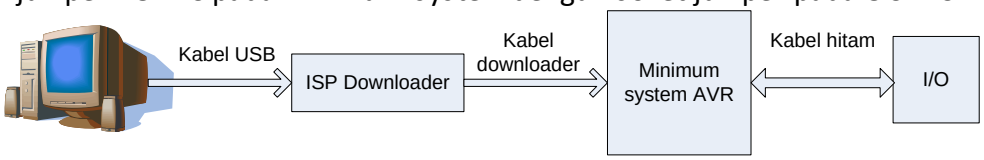
ALAT DAN BAHAN YANG DIGUNAKAN

- 1 set PC/Laptop yang sudah berisi program Code Vision dan Khazama
- 1 buah multimeter dengan fasilitas frequency counter
- 1 buah ISP Downloader AVR
- 1 buah sistem minimum AVR
- 1 buah I/O
- 1 buah kabel printer USB
- 2 buah kabel pita hitam

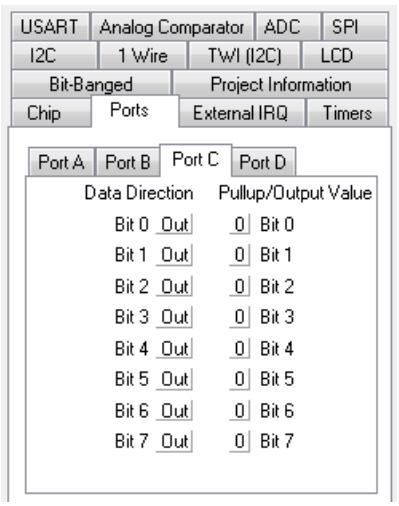
PROSEDUR

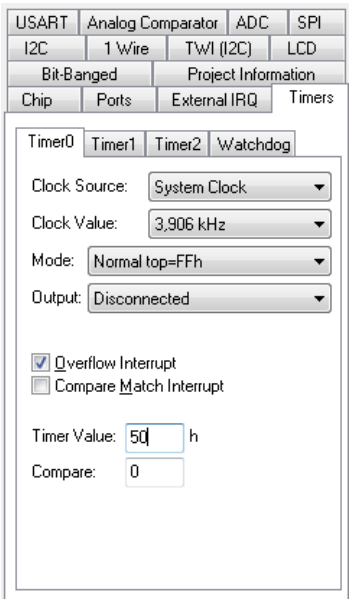
TIMER

1. Rangkailah peralatan yang diperlukan seperti gambar dibawah. Hubungkan soket jumper PORTC pada minimum system dengan soket jumper pada OUTPUT Trainer I/O.



2. Buka program Code Vision AVR
3. Buatlah project baru. Pada saat mengeset chip dan clock, set juga bagian PORTC untuk LED serta Timer seperti gambar dibawah. Kemudian simpanlah file tersebut.
   - Nilai Clock Value adalah Nilai Clock Frequency Chip (F_{clock}) dibagi dengan prescaller dimana prescaller tersebuat bernilai 1, 8, 64, 256, atau 1024
   - Contoh jika nilai F_{clock} = 4 Mhz maka jika kita ingin menggunakan prescaller 1024 setting Clock Value nya adalah 4Mhz/1024=3,906 Khz





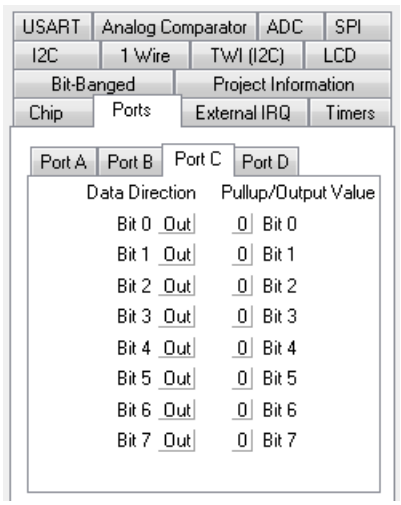
4. Perhatikan blok program berikut.

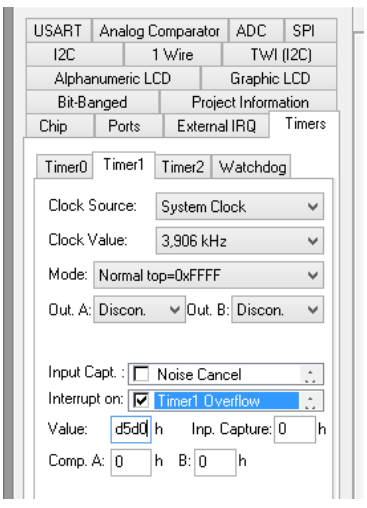
```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x05;
TCNT0=0x50;
OCR0=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;

// Global enable interrupts
#asm("sei")
```

5. Tuliskan script berikut dalam interrupt:

```
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Reinitialize Timer 0 value
TCNT0=0x50;
// Place your code here
if (PINC==0)
  {PORTC=0xFF;}
else
  {PORTC=0x00;}
}
```

6. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khanzama AVR Programer. Klik auto program.
7. Perhatikan dan catat nyala LED.
8. Ubah nilai TCCR0 menjadi 0x04 dan 0x03.
9. Perhatikan perbedaan nyala LED dengan nilai TCCR yang berbeda-beda.
10. Ulangi langkah 3-9 untuk Timer2 dengan TCNT=0x80;
11. Buat Project baru untuk Timer1 dengan TCNT = 0xd5d0





12. Perhatikan blok berikut

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 3,906 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x05;
TCNT1H=0xD5;
TCNT1L=0xD0;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x04;

// Global enable interrupts
#asm("sei")
```

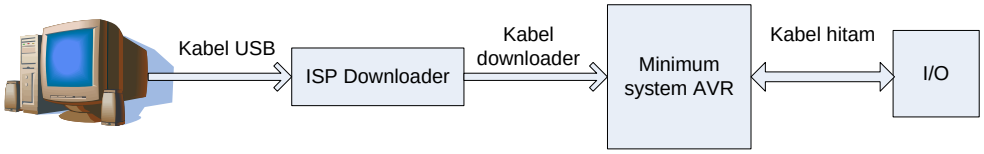
13. Tuliskan script berikut dalam blok interrupt

```
// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer1 value
TCNT1H=0xD5D0 >> 8;
TCNT1L=0xD5D0 & 0xff;
// Place your code here
if (PINC==0)
  {PORTC=0xFF;}
else
  {PORTC=0x00;}
}
```

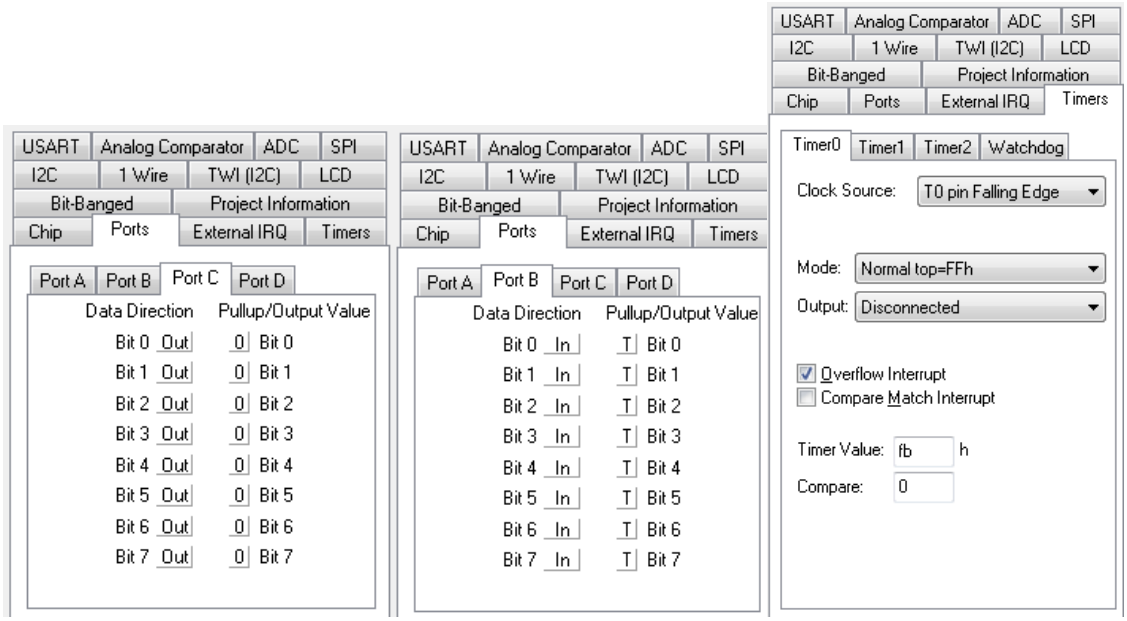
14. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khanzama AVR Programer. Klik auto program.
15. Perhatikan dan catat nyala LED.
16. Ubah nilai TCCR1B sehingga menjadi 0x04, 0x03, 0x02, 0x01.
17. Perhatikan perbedaan nyala LED dengan nilai TCCR yang berbeda-beda.
18. Buatlah program menggunakan Timer 16 bit (Timer1) dengan nilai TCCR1B=0x05, Ubah nilai TCNT sehingga menghasilkan timer 1 detik.

COUNTER

1. Rangkailah peralatan yang diperlukan seperti dalam gambar dibawah. Hubungkan soket jumper PORTC pada minimum system dengan soket jumper pada OUTPUT LED. Hubungkan kaki IS1 ke PORTB.0 (T0).



2. Buka program Code Vision AVR
3. Buatlah project baru. Pada saat mengeset chip dan clock, set juga bagian PORTC untuk LED serta Timer seperti gambar dibawah. Kemudian simpanlah file tersebut.



4. Perhatikan blok program berikut.

```
// Timer/Counter 0 initialization
// Clock source: T0 pin Falling Edge
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x06;
TCNT0=0xFB;
OCR0=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;

// Global enable interrupts
#asm("sei")
```

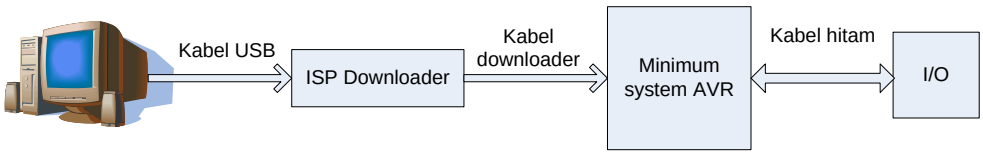
5. Tuliskan script berikut dalam interrupt:

```
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Reinitialize Timer 0 value
TCNT0=0xFB;
// Place your code here
if (PINC==0)
  {PORTC=0xFF;}
else
  {PORTC=0x00;}
}
```

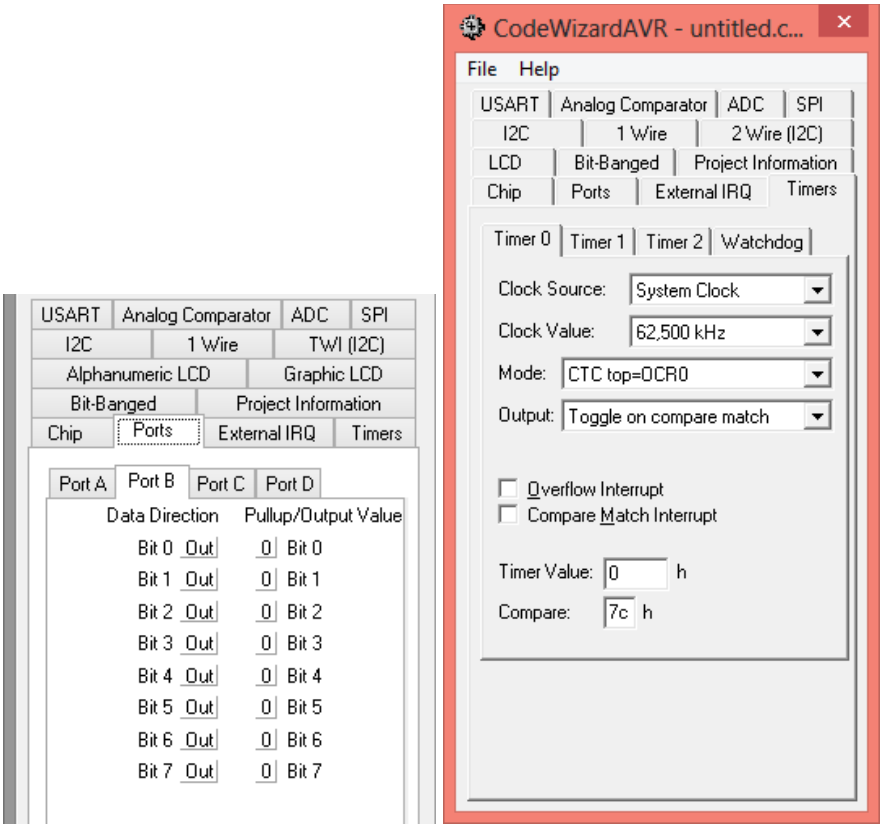
6. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khanzama AVR Programer. Klik auto program.
7. Tekan push button IS1 Perhatikan dan catat nyala LED.
8. Lakukan hal serupa dengan Timer dengan mengganti nilai TCNT0 menjadi 0xFC dan 0xF0 (lihat tabel counter pada data hasil percobaan).

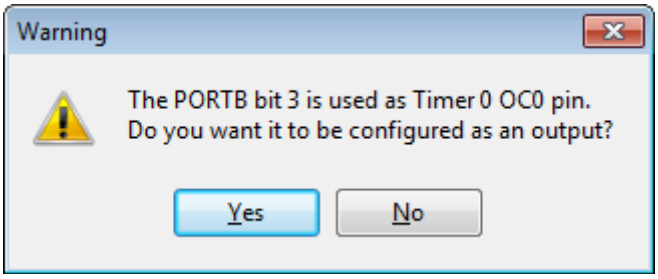
CTC

1. Rangkailah peralatan yang diperlukan seperti dalam Gambar 1.2. Hubungkan PORTB pada minimum system dengan kabel data pada OUTPUT TRAINER.



2. Buka program Code Vision AVR
3. Buatlah project baru. Pada saat mengeset chip dan clock, set juga bagian PORTB untuk LED serta Timer. Kemudian simpanlah file tersebut.





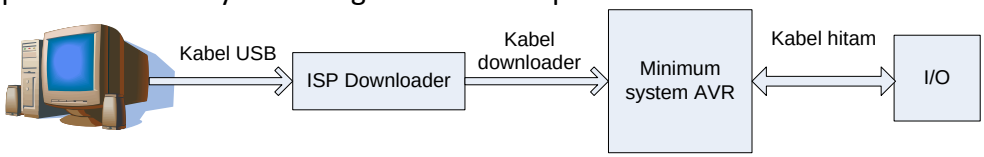
4. Klik Yes jika muncul dialog diatas
5. Perhatikan blok program berikut.

```
// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out
// Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0
// State1=0 State0=0
PORTB=0x00;
DDRB=0xFF;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 62,500 kHz
// Mode: CTC top=OCR0
// OC0 output: Toggle on compare match
TCCR0=0x1B;
TCNT0=0x00;
OCR0=0x7C;
```

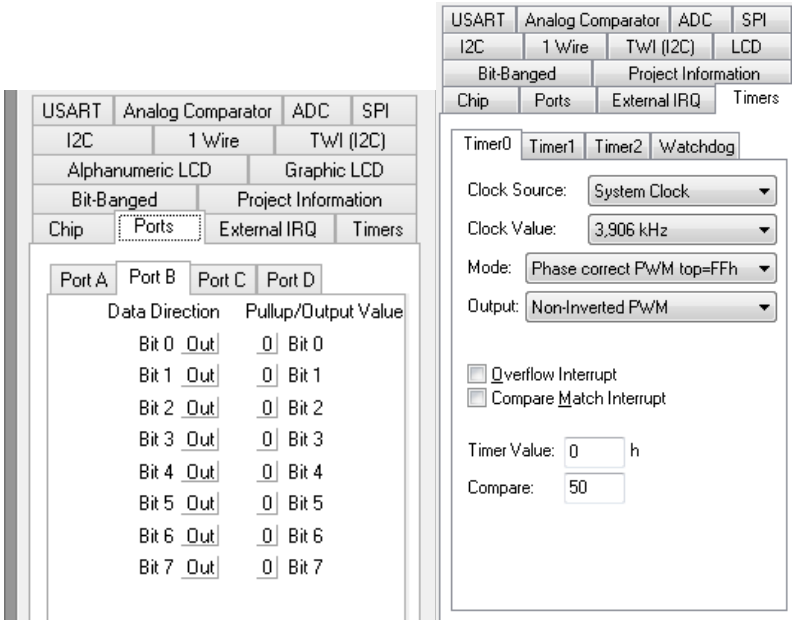
6. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khanzama AVR Programer. Klik auto program.
7. Lihat nyala LED di PORTB.3.
8. Ukur frekuensi pada PORTB.3 dengan frequency counter di AVOmeter
9. Ubah Nilai OCR0 menjadi 0x3E dan 0x1E (lihat tabel CTC pada data hasil percobaan)
10. Lihat nyala LED di PORTB.3.
11. Ukur frekuensi pada PORTB.3 dengan frequency counter di AVOmeter
12. Amati Nyala LED
13. Buatlah program yang dapat membangkitkan sinyal dengan frekuensi 1 Hz.

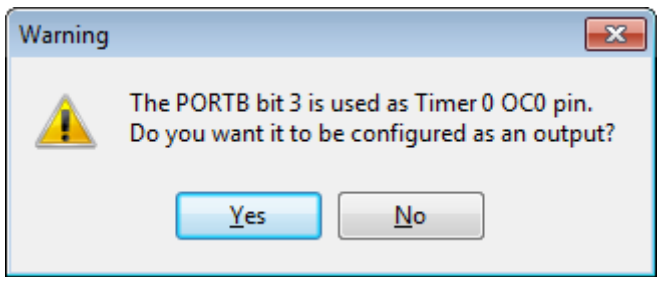
PULSE WIDTH MODULATION (PWM)

1. Rangkailah peralatan yang diperlukan seperti dalam Gambar 1.2. Hubungkan PORTB pada minimum system dengan kabel data pada OUTPUT TRAINER.



2. Buka program Code Vision AVR
3. Buatlah project baru. Pada saat mengeset chip dan clock, set juga bagian PORTB untuk LED serta Timer. Kemudian simpanlah file tersebut.





4. Klik Yes jika muncul dialog diatas
5. Perhatikan blok program berikut.

```
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=0 State2=T State1=T State0=T
PORTB=0x00;
DDRB=0xFF;


// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 3,906 kHz
// Mode: Phase correct PWM top=0xFF
// OC0 output: Non-Inverted PWM
TCCR0=0x65;
TCNT0=0x00;
OCR0=0x50;
```

6. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khanzama AVR Programer. Klik auto program.
7. Lihat nyala LED di PORTB.3.
8. Ukur frekuensi pada PORTB.3
9. Ubah Nilai OCR0 menjadi 0x00 dan 0xFF (lihat tabel PWM pada data hasil percobaan)
10. Amati Nyala LED
11. Ulang langkah 3-9 dengan mode Fast PWM.
12. Buatlah program yang dapat mengubah nilai Duty Cycle sesuai dengan input Potensiometer !
    Potensiometer 0 V (minimum) → Duty Cycle 0%
    Potensiomter 5V (Maksimum) → Duty Cycle 100%

DATA HASIL PERCOBAAN

TIMER

No	TIMER	TCCRx	TCNTx	Kondisi Led	Delay
1	Timer0	0x05	0x50		
2	Timer0	0x04	0x50		
3	Timer0	0x03	0x50		
6	Timer2	0x05	0x80		
7	Timer1	0x05	0xD5D0		
8	Timer1	0x04	0xD5D0		
9	Timer1	0x03	0xD5D0		
10	Timer1	0x02	0xD5D0		
11	Timer1	0x01	0xD5D0		
12	Timer1			00000000 → 11111111	1 s

Counter

No	TIMER	TCCRx	TCNTx	Jumlah Penekanan Tombol IS1 Hingga LED Menyala	Jumlah Penekanan Tombol IS1 Hingga LED Mati
1	Timer0	0x06	0xFB		
2	Timer0	0x06	0xFC		
3	Timer0	0x06	0xF0		

CTC

Nilai OCR0/OCR1A/OCR1B/OCR2	N	Nyala LED	Frekuensi
0x7C			
0x3E			
0x1E			
			1 Hz

PWM

Fast PWM			
Nilai OCR0	Nyala LED	Tegangan LED	Frekuensi
0x00			
0x50			
0xFF			

Phase Correct PWM			
Nilai OCR0	Nyala LED	Tegangan LED	Frekuensi
0x00			
0x50			
0xFF			

ANALISIS DATA

TIMER

1. Analisis data hasil pada tabel Timer dengan menggunakan rumus :

$$T_{timer0} = \frac{\left((1+255) - TCNT0\right) \cdot Prescaler}{f_{clock}}$$

$$T_{timer1} = \frac{\left((1+65535) - TCNT1\right) \cdot Prescaler}{f_{clock}}$$

Hitung Nilai T_{timer} :

TIMER	TCCRx	TCNTx	Fclock (Hz)	Prescaler	8 / 16 bit	T (s)	Keterangan
Timer0	0x05	0x50	4.000.000	1.024	255	0,05	Delay tidak terlihat
Timer0	0x04	0x50					
Timer0	0x03	0x50					
Timer2	0x05	0x80					
Timer1	0x05	0xD5D0					
Timer1	0x04	0xD5D0					

Timer1	0x03	0xD5D0					
Timer1	0x02	0xD5D0					
Timer1	0x01	0xD5D0					

2. Bagaimana anda mendapatkan timer dengan T_{timer} 1 detik dengan menggunakan Timer1?

COUNTER

1. Analisis data hasil pada tabel counter

CTC

1. Hitung nilai frekuensi

Nilai OCR0/OCR1A/OCR1B/OCR2	N	Nyala LED	Frekuensi
0x7C			
0x3E			
0x1E			

2. Bagaimana cara anda membangkitkan sinyal dengan frekuensi 1 detik menggunakan Timer/Counter mode CTC?

PWM

1. Hitung nilai frekuensi dan duty cycle

Fast PWM			
Nilai OCR0	Duty Cycle Berdasarkan Nilai OCR0	Duty Cycle Berdasarkan Nilai Tegangan LED	Frekuensi
0x00			
0x50			
0xFF			

Phase Correct PWM			
Nilai OCR0	Duty Cycle Berdasarkan Nilai OCR0	Duty Cycle Berdasarkan Nilai Tegangan LED	Frekuensi
0x00			
0x50			
0xFF			

2. Jelaskan perbedaan mode Fast PWM dengan Phase Correct PWM
3. Mengapa nyala led pada mode Fast PWM berkedip lebih cepat daripada mode Phase Correct PWM?
4. Source Code Program langkah no 11.