

MODUL 5

TIMER DAN INTERRUPT

Timer & Counter merupakan fitur yang telah tertanam di mikrokontroler AVR yang memiliki fungsi terhadap waktu. Fungsi pewaktu yang dimaksud disini adalah penentuan kapan program tersebut dijalankan, tidak hanya itu saja fungsi timer yang lainnya adalah PWM, ADC, dan Oscillator. Prinsip kerja timer dengan cara membagi frekuensi (prescaler) pada clock yang terdapat pada mikrokontroler sehingga timer dapat berjalan sesuai dengan frekuensi yang di kehendaki.

Timer merupakan fungsi waktu yang sumber clocknya berasal dari clock internal. Sedangkan counter merupakan fungsi perhitungan yang sumber clocknya berasal dari external mikrokontroler. Salah satu contoh penggunaan fungsi timer yaitu pada jam digital yang sumber clocknya bisa menggunakan crystal oscillator dan contoh penggunaan counter pada penghitung barang pada konveyor yang sumber clocknya berasal dari sensor yang mendeteksi barang tersebut.

Pada Arduino AVR yang digunakan yaitu Atmel AVR Atmega 168 atau Atmega328. Kedua chip tersebut kompatibel tetapi memiliki memori internal yang berbeda, walaupun keduanya memiliki 3 timer yang disebut dengan Timer0, Timer1, dan Timer2. Timer0 dan Timer2 memiliki waktu 8 bit, sedangkan untuk Timer1 memiliki waktu 16 bit. Antara 8 bit dengan 16 bit hanya berbeda pada resolusi waktu, karena 8 bit hanya memiliki 256 nilai, sedangkan 16 bit memiliki 65536 nilai yang berarti resolusi waktu lebih tinggi.

Timer0

Timer0 yaitu timer 8 bit. Pada Arduino timer ini digunakan untuk fungsi timer seperti delay(), millis(), dan micros(). Selain itu terdapat timer0 register dimana dapat diubah, akan tetapi timer0 register mempengaruhi fungsi timer Arduino.

Timer1

Timer1 yaitu timer 16 bit. Pada Arduino Uno timer1 biasa digunakan untuk library Servo, sedangkan untuk Arduino Mega pada timer5.

Timer2

Timer2 yaitu timer 8 bit, seperti timer0. Timer2 biasa digunakan untuk fungsi nada atau tone().

Timer3, Timer4, Timer5

Timer 3, 4, 5 hanya tersedia pada Arduino mega dimana semua Timer memiliki waktu 16 bit.

Timer Register

Timer register mempengaruhi Timer yang lainnya, timer ini dapat diubah atau dikonfigurasi.



- TCCR_x** - Timer/Counter register. Berfungsi untuk mengatur Prescale.
- TCNT_x** - Timer/Counter register. Untuk menyimpan nilai timer.
- OCR_x** - Keluaran pembanding register
- ICR_x** - Masukan untuk register (hanya terdapat pada timer 16 bit)
- TIMSK_x** - Timer/Counter untuk mengaktifkan/menon-aktifkan interupsi timer
- TIFR_x** - Timer/Counter untuk menunjukkan timer interupsi yang tertunda

Interupsi

Interupsi merupakan kondisi dimana terdapat masukan pada saat program berjalan. Sebuah interupsi dapat langsung dijalankan ketika terdapat kondisi interupsi khusus. Jika interupsi dijalankan maka program yang sedang berjalan akan menunggu sampai interupsi tersebut selesai dikerjakan.

Interupsi pada umumnya dapat diaktifkan/ dinonaktifkan dengan cara interrupts()/noInterrupts(). Secara default pada Arduino interupsi diaktifkan, penanda interupsi diaktifkan/dinonaktifkan dengan pengaturan bit pada TIMSK_x (The Interrupt Mask Register).

Ketika terjadi interupsi, maka interupsi akan ditetapkan pada TIFR_x (The Interrupt Flag Register) sebagai penanda dari interupsi. Jika menggunakan ISR (Interrupt Service Routines) setelah tidak dipakai maka akan terhapus secara otomatis atau dapat juga dihapus secara manual pada TIFR_x.

Pada Arduino Uno dan Arduino Nano memiliki 2 PIN interrupt yaitu INT0 pada PIN 2 digital dan INT1 pada PIN 3 digital.

Untuk membuat interupsi pada Arduino dapat menggunakan fungsi berikut.

```
attachInterrupt(interrupt, function, mode);
```

Pada bagian function dapat diisi dengan program yang ingin dijalankan saat PIN interrupt dipicu. Kemudian, terdapat 4 mode yang dapat memicu interupsi, diantaranya :

- a. Mode LOW, pada mode ini interrupt akan diaktifkan saat PIN INT memiliki logika LOW dan akan aktif selama masih berlogika LOW.
- b. Mode CHANGE, pada mode ini interrupt akan diaktifkan saat terjadi perubahan logika baik dari LOW ke HIGH ataupun HIGH ke LOW. Jalannya program hanya bersifat sementara, ketika program pada interrupt selesai maka akan melanjutkan program sebelumnya.
- c. Mode RISING, pada mode ini sama seperti mode CHANGE yaitu interrupt akan aktif ketika terjadi perubahan logika dari LOW ke HIGH, dan akan kembali ke program sebelumnya setelah selesai proses interrupt.
- d. Mode FALLING, pada mode ini juga sama seperti mode CHANGE yaitu interrupt akan aktif ketika terjadi perubahan logika dari HIGH ke LOW, dan juga akan kembali ke program sebelumnya setelah selesai proses interrupt.

PWM dan Timer

Hubungan antara timer dengan PWM sangatlah terlihat, jika kita lihat dari datasheet terdapat PIN yang memiliki data OCRxA, OCRxB, atau OCRxC dimana x berarti interval waktu antara 0 sampai dengan 5. Fungsi PWM saling berkaitan dengan fungsi PIN yang lainnya.

Arduino UNO memiliki 3 Timer dan 6 PIN keluaran PWM. Diantaranya yaitu,

- PIN 5 dan 6, dikendalikan oleh Timer0
- PIN 9 dan 10, dikendalikan oleh Timer1
- PIN 11 dan 3, dikendalikan oleh Timer2

Penggunaan Timer Overflow Untuk LED Blinking

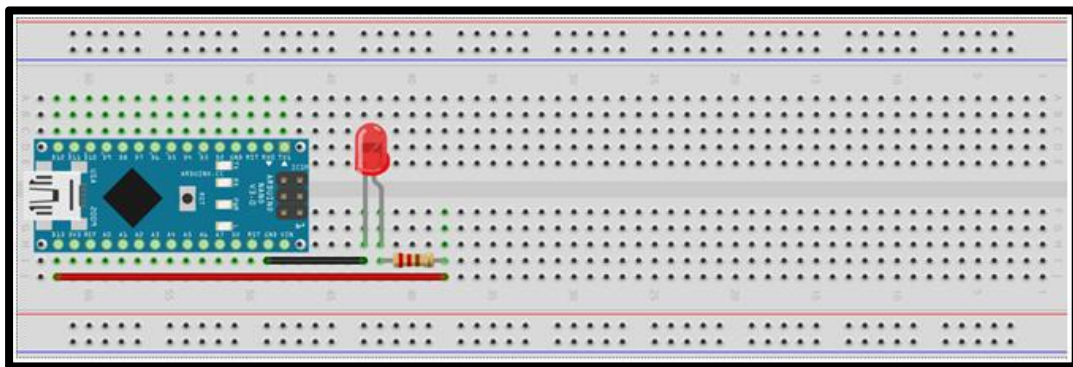
Penggunaan Timer untuk membuat led berkedip setiap 500 milidetik. Pada dasarnya LED akan berubah kondisi setiap terjadi interupsi timer overflow. Nilai

register timer di-set sedemikian rupa sehingga menghasilkan waktu overflow setiap 500 milidetik.

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. LED
4. Resistor
5. Kabel Jumper

b. RANGKAIAN DAN LANGKAH PRAKTIKUM



Gambar 5.1. Rangkaian LED Timer Overflow

```
boolean dataLed = 0;
unsigned int overflowCount = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 34286;
  TCCR1B |= (1 << CS12); //256 prescaler
  TIMSK1 |= (1 << TOIE1);
  sei();
}

ISR(TIMER1_OVR_vect)
{
  TCNT1 = 34286;
  if(dataLed == 0){
    dataLed = 1;
  }else{
    dataLed = 0;
  }
  digitalWrite(13, dataLed);
  overflowCount++;
}
```

```

}

void loop()
{
  Serial.print("Overflow count = ");
  Serial.println(overflowCount, DEC);
  delay(100);
}

```

Modul 5.1. Program Timer Overflow

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasangkan kabel jumper ke pin 13, lalu ujung lainnya sambungkan ke resistor.
4. Lalu pasang LED bagian yang panjang ke resistor.
5. Pasang kabel jumper ke ground pasangkan ke bagian pendek LED.
6. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
7. Amati apa yang terjadi.

Counter Mode Falling Edge

Program ini dibuat untuk mengetahui banyaknya penekanan switch atau button pada input counter. Konfigurasi counter menggunakan mode falling edge.

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board

b. LANGKAH PRAKTIKUM

```

void setup()
{
  Serial.begin(9600);
  TCCR1A=0x00;
  TCCR1B=0x07;
  TCNT1H=0x00;
  TCNT1L=0x00;
  ICR1H=0x00;
  ICR1L=0x00;
  OCR1AH=0x00;
  OCR1AL=0x00;
  OCR1BH=0x00;
  OCR1BL=0x00;
}

void loop()
{
  Serial.print("Nilai TCNT1 = ");

```

```
Serial.println(TCNT1, DEC);  
delay(100);  
}
```

Modul 5.2. Program Counter Falling Edge

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
4. Amati apa yang terjadi pada Serial Monitor.

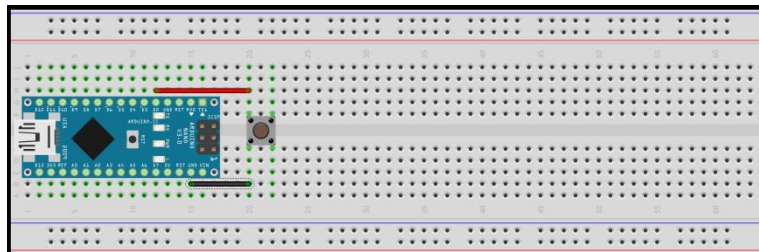
Interupsi Input

Program berikut untuk memberikan contoh penggunaan interupsi untuk input.

a. ALAT DAN BAHAN

1. Arduino Nano
2. Project Board
3. Push Button
4. Kabel Jumper

b. LANGKAH PRAKTIKUM



Gambar 5.2. Rangkaian LED Timer Overflow

```
unsigned long counts;  
unsigned long previousMillis;  
  
void impulse() { // dipanggil setiap ada sinyal FALLING  
di pin 2  
    counts++;  
}  
  
#define LOG_PERIOD 1000 // cetak tiap detik  
  
void setup() { //setup subprocedure  
    counts = 0;  
    Serial.begin(9600);  
    pinMode(2, INPUT);  
    digitalWrite(2, HIGH); // mengaktifkan internal pull up  
    //define external interrupts
```

```
attachInterrupt(digitalPinToInterrupt(2), impulse,
FALLING);
Serial.println("Start counter");
}

void loop() { //main cycle
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis > LOG_PERIOD) {
        previousMillis = currentMillis;
        Serial.println(counts);
        counts = 0;
    }
}
```

Modul 5.3. Program Interupsi dengan Push Button

Langkah – langkahnya :

1. Tulis program diatas di sketch arduino.
2. Siapkan alat dan bahan yang disebutkan diatas.
3. Pasang Push button pada breadboard.
4. Hubungkan kabel jumper dari PIN 2 digital ke push button.
5. Hubungkan kabel jumper dari push button ke ground (GND).
6. Pasang kabel usb ke arduino dan PC, lalu upload program ke arduino.
7. Amati apa yang terjadi pada Serial Monitor.