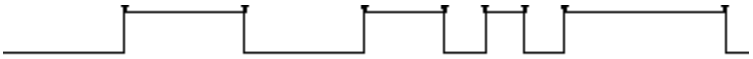
Teknik PWM pada Sistem Minimum ATMega128 DST-128 AVR STAMP

Setelah pada artikel sebelumnya membahas penggunaan *uart* dan *timer* pada ATMEGA 128, sekarang mencoba fungsi lain dari *timer* yaitu PWM (Pulse Width Modulator). PWM berguna untuk mengatur kecepatan motor DC dan juga untuk mengatur kecerahan led, tetapi fungsi PWM tidak hanya itu.

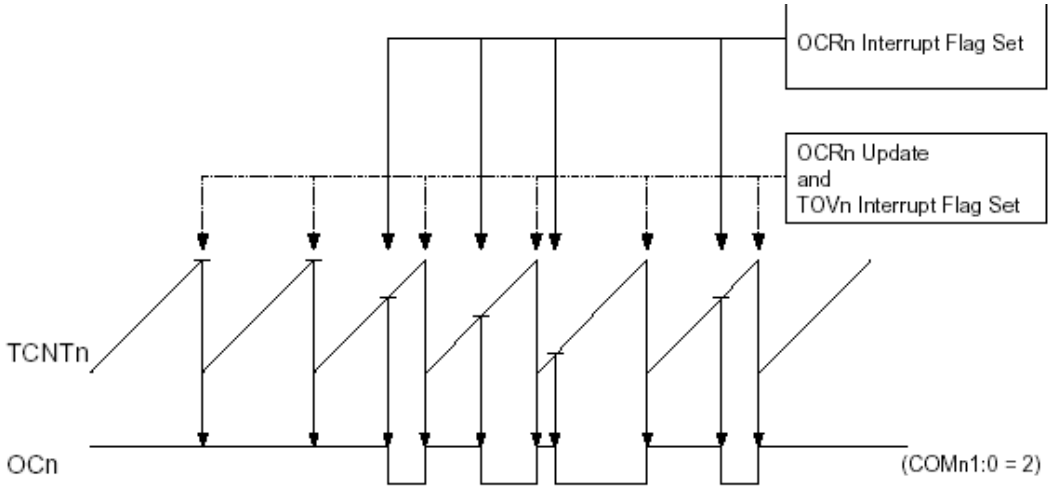
Pada prinsipnya PWM itu adalah mengatur lamanya penyalaan alat tersebut dalam satu satuan waktu yang sangat cepat, diatas 20Khz.



Gb 1.

Gambar diatas adalah contoh PWM, sehingga jika lebar pulsa pada saat ‘1’ dirubah-rubah dan jika dirata-rata tegangan yang dihasilkan juga akan berubah-ubah sesuai dengan lebar pulsa pada saat ‘1’.

Untuk dapat menghasilkan gelombang seperti diatas dapat dilakukan dengan menggunakan *timer* yang ada pada mikrokontroler. Pada ATMEGA 128 tidak direpotkan untuk pembuatan program penghasil gelombang PWM ini, cukup dengan mengatur nilai dari sebuah register maka sudah dapat mengatur PWM dan membangkitkannya pada sebuah PORT mikrokontroler seperti yang ditunjukkan oleh gambar dibawah ini.



Gb 2.

Register yang perlu diatur adalah OCRn dimana n adalah *timer* yang digunakan untuk pembangkit PWM. Register TCNTn adalah register nilai *timer* pada mode ini *timer* akan menghitung naik dan nilainya dibandingkan dengan nilai OCRn jika nilainya sama maka *pin* OCn akan dibuat ‘0’. Dengan demikian jika mengubah-ubah nilai OCRn maka lamanya ‘0’ juga ikut berubah-ubah.

Setelah memahami prinsipnya selanjutnya mengatur register-register yang digunakan supaya timer dapat dioperasikan sebagai PWM.

Register-registernya adalah :

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

1. Bit FOC0 berfungsi jika timer tidak dioperasikan sebagai PWM,karena artikel kali ini adalah PWM maka fungsi dari FOC0 ini tidak perlu diatur.
2. Bit WGM00 dan WGM01 berfungsi untuk mengatur fungsi timer. Jika 00 maka timer akan beroperasi dalam mode normal, jika 01 timer beroperasi mode PWM phase correct, jika 10 maka timer beroperasi mode CTC, jika 11 timer beroperasi mode fast PWM.
3. Bit COM00 dan COM01 berfungsi untuk menentukan keadaan pin OCn pada saat compare match terjadi
4. Bit CS00,CS01 dan CS02 berfungsi untuk menentukan besarnya frekuensi yang digunakan timer.

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	AS0	TCN0UB	OCR0UB	TCR0UB	ASSR
Read/Write	R	R	R	R	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

1. Bit AS0 untuk menentukan sumber clock yang digunakan. Jika ‘0’ maka timer menggunakan system clock yang bekerja pada mikrokontroler. Jika ‘1’ maka timer bekerja menggunakan clock yang berasal dari pin TOSC1.
2. Bit TCN0UB untuk menandakan register TCNT0 siap menerima nilai yang akan diisikan.
3. Bit OCR0UB sama seperti bit TCN0UB, yang berbeda adalah register yang disikan adalah OCR0.

4. Bit TCR0UB sama seperti bit sebelumnya, registernya adalah TCCR0.

Hanya dengan mengatur kedua register diatas sudah dapat membangkitkan PWM pada pin OC0 dengan terlebih dahulu mengatut port direction menjadi output.

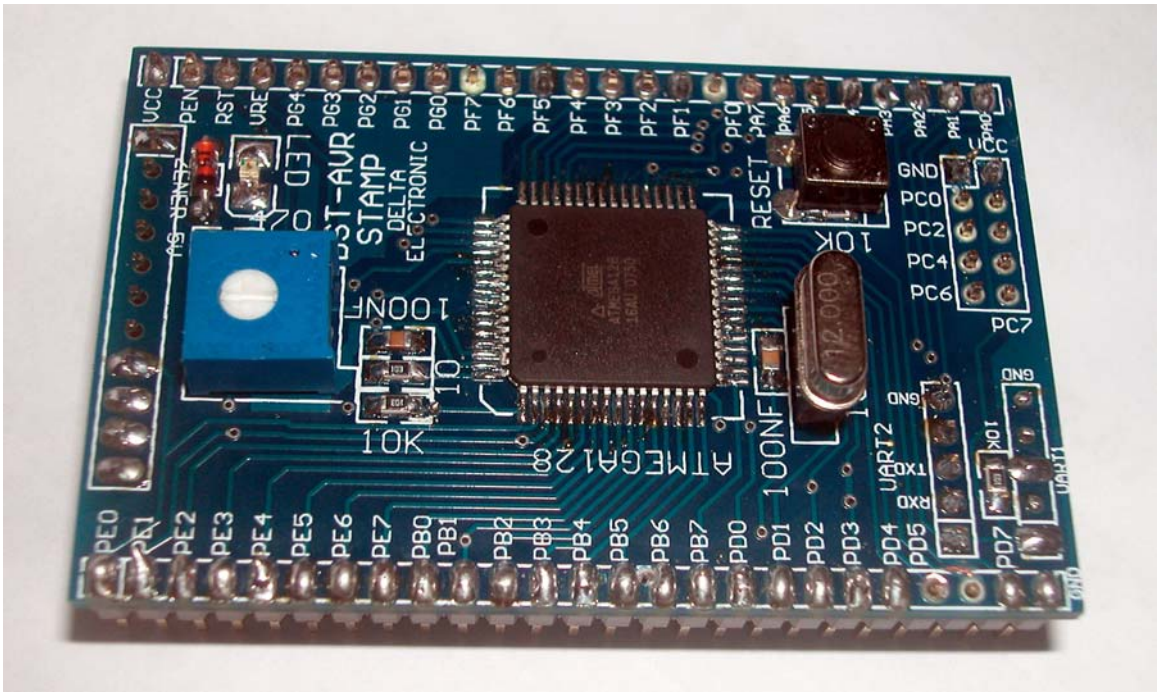
Aplikasi pada artikel ini adalah mencoba mengatur kecerahan led menggunakan teknik PWM. Led dihubungkan pada pin OC0. Potongan program dibawah ini berfungsi untuk mengatur register timer 0 bekerja pada mode fast PWM dengan frekuensi 11,719Khz.

```
1: // Timer/Counter 0 initialization
2: // Clock source: System Clock
3: // Clock value: 11,719 kHz
4: // Mode: Fast PWM top=FFh
5: // OC0 output: Non-Inverted PWM
6: ASSR=0x00;
7: TCCR0=0x6F;
8: TCNT0=0x00;
9: OCR0=0x00;
```

Potongan program ini dijalankan pertama kali setiap mikrokontroler reset. Karena setiap kali mikrokontroler reset maka register-register diatas kembali dalam keadaan default. Lihat penjelasan pada register.

Setelah mengatur timer maka program utamanya adalah mengatur nilai OCR0 untuk mengatur kecerahan led. Potongan program ini dilakukan terus menerus oleh mikrokontroler.

```
while(1)
      {
            for(i=0;i<255;i++)
            {
                OCR0=i;
                _delay_ms(1);
            }
            for(i=255;i>0;i--)
            {
                OCR0=i;
                _delay_ms(1);
            }
      }
```



Nilai OCR0 berubah sesuai dengan nilai I, dimana nilai I sendiri naik turun dari 0-255, sehingga led akan menyala terang-redup-terang dan seterusnya. Program ini juga dapat dibuat untuk mengatur kecepatan motor dengan menaikkan frekuensi PWM dan menambahkan sebuah transistor power untuk mendrive motor. Aplikasi dapat didownload di AN0153, www.delta-electronic.com bagian application note **Januar, Delta Electronic.com**