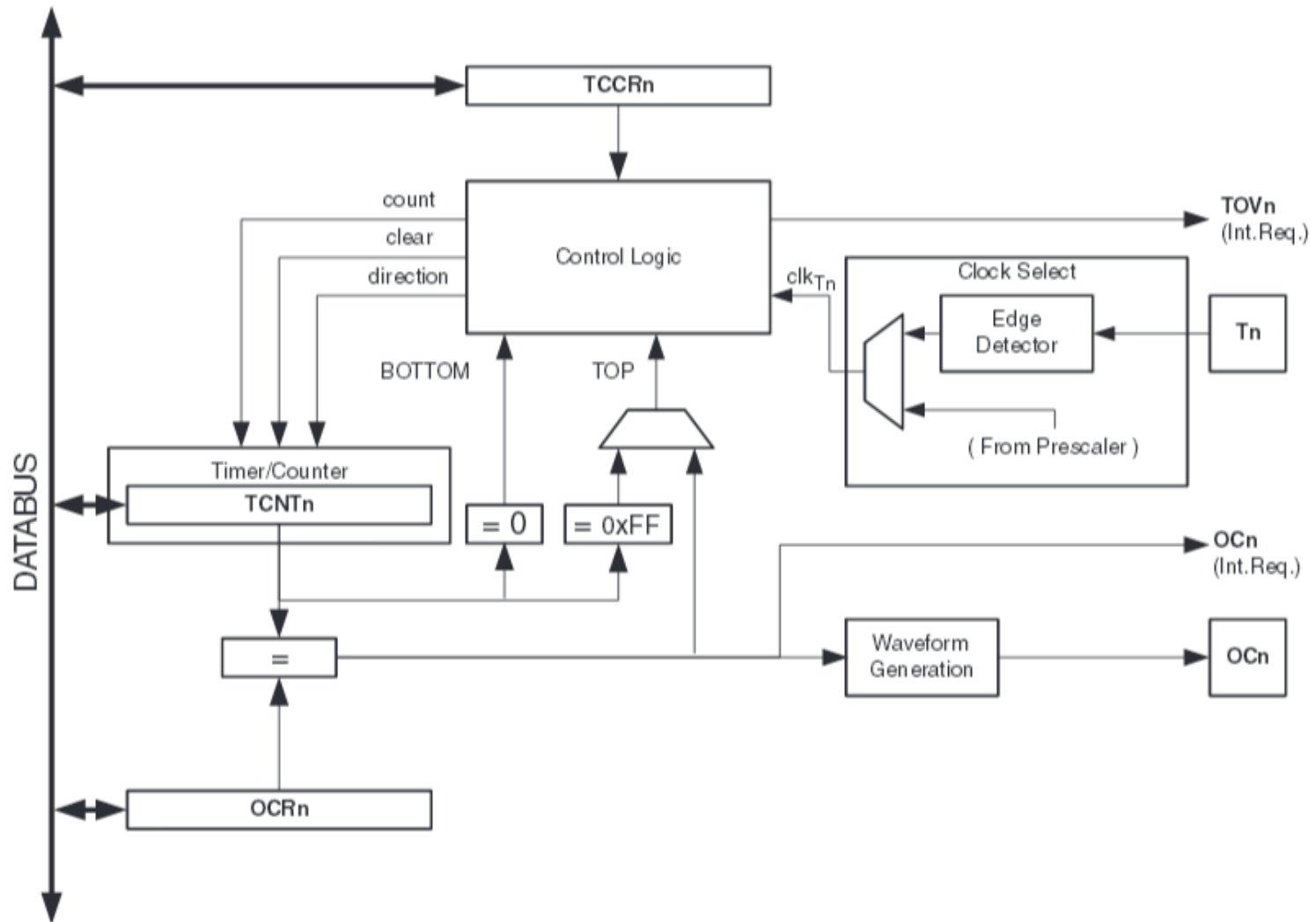


# Timery i PWM

Mariusz Wałdoch

# Budowa timera 0

Figure 27. 8-bit Timer/Counter Block Diagram

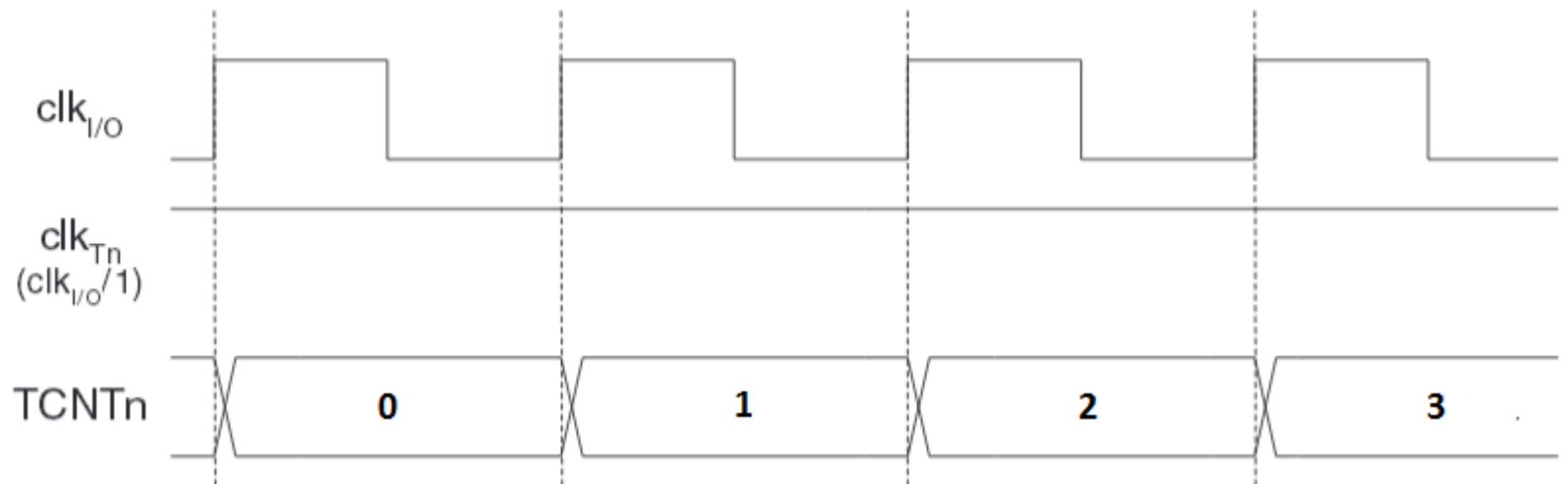


# Budowa timera 0

- TCCR0 – Timer/Counter Control Register
- TCNT0 – Timer/Counter Register
- OCR0 – Output Compare Register
- TIMSK – Timer/Counter Interrupt Mask Register
- TIFR – Timer/Counter Interrupt Flag Register

# Jak działa timer?

**Figure 34.** Timer/Counter Timing Diagram, no Prescaling

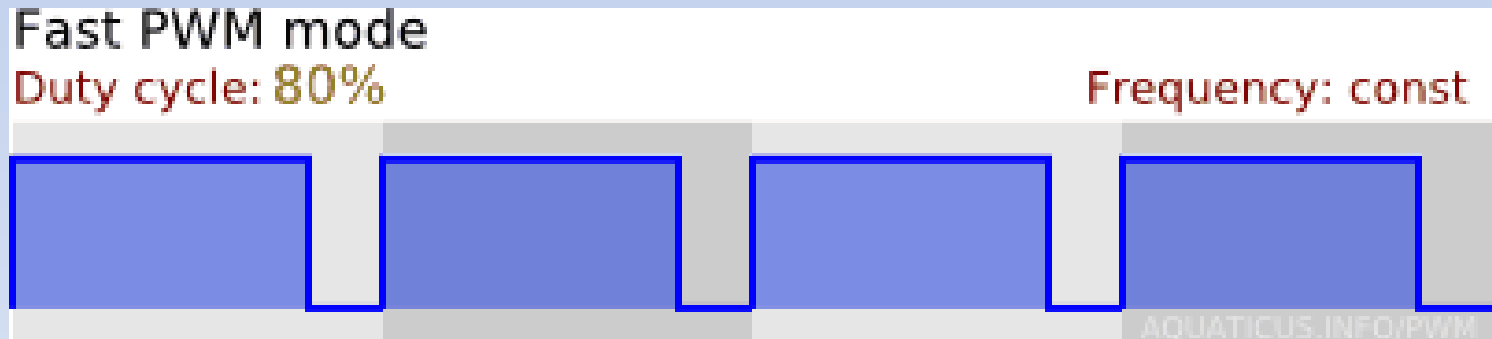


# Co jest „programowalne”?

- Zegar wejściowy – można dzielić poprzez wbudowany preskaler (typowe wartości dzielnika: 8, 64, 256, 1024)
- Wartość przy której ma nastąpić zerowanie licznika i powtarzanie cyklu (OCRn)
- Źródło zegara – zegar wewnętrzny mikrokontrolera lub zewnętrzny sygnał podany na odpowiedni pin mikrokontrolera
- Działanie pinów powiązanych z układem timera (Input Capture, wyprowadzenia PWM)

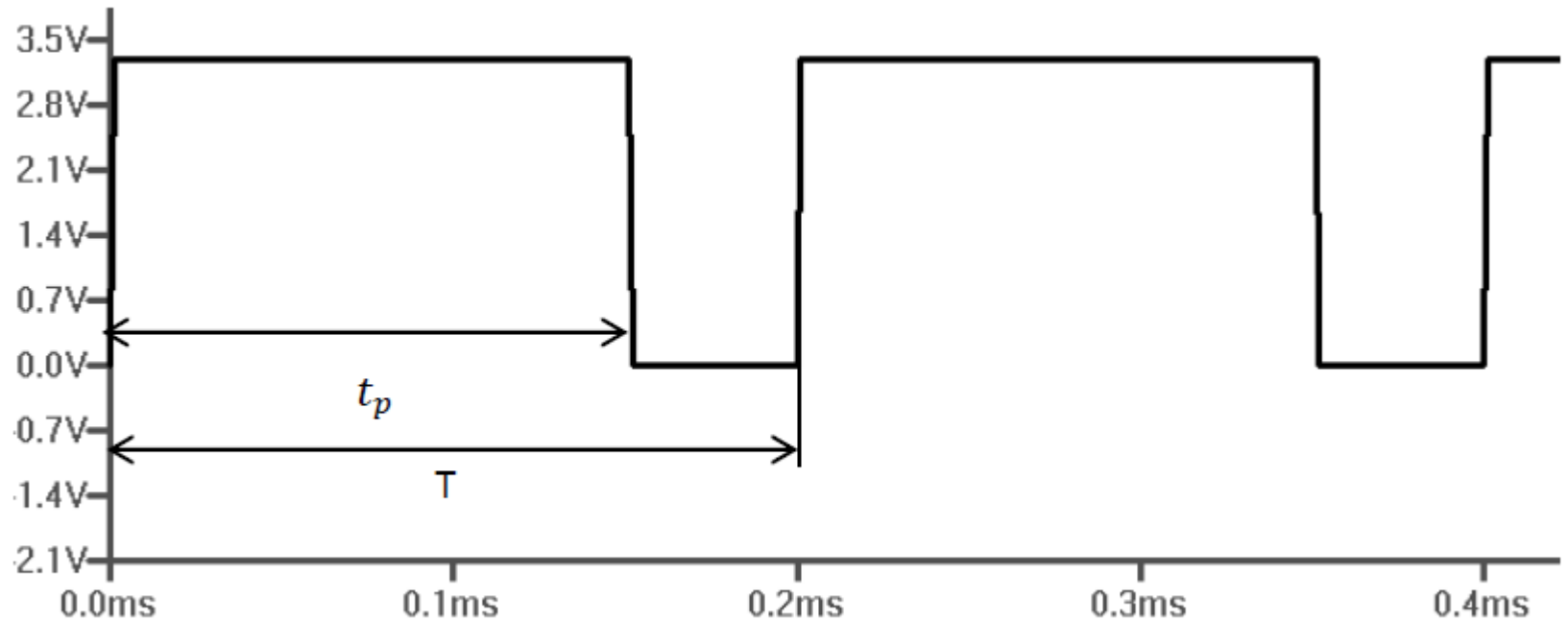
# PWM – co to jest?

- PWM – Pulse Width Modulation – modulacja szerokości impulsu. Pozwala na proste sterowanie jasnością diod LED czy prędkością obrotową silników przy użyciu mostków H.



Źródło: [aquaticus.info](http://aquaticus.info)

# PWM – co to jest?



Przebieg sygnału PWM o wypełnieniu 75%.

# Tryby pracy timera0

- Normal mode – w momencie przepełnienia licznika może być wygenerowane przerwanie.
- CTC – licznik będzie zerowany przy zgodności TCNT0 i OCR0. OCR0 definiuje rozdzielczość timera. Daje możliwość generowania przebiegów o zmiennej częstotliwości. Można również zmieniać stan linii OC0 mikrokontrolera przy zgodności wymienionych wcześniej rejestrów.

$$f_{OCn} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRn)}$$

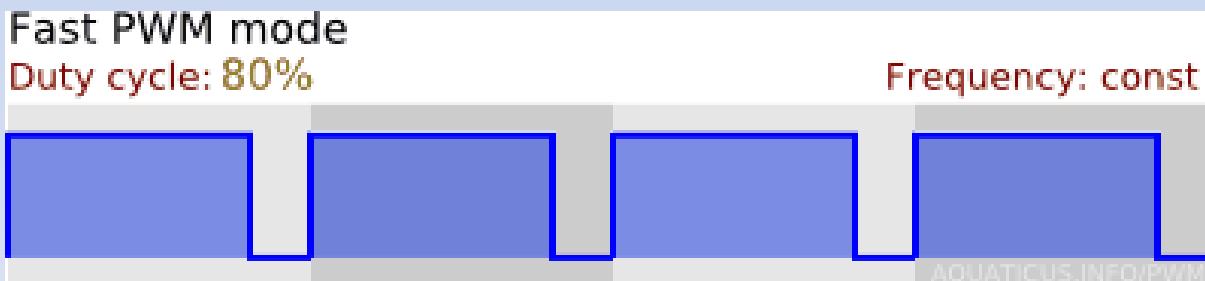
- $f_{clk\_I/O}$  – zegar Atmegi
- $N$  – preskaler (dzielnik zegara  $f_{clk\_I/O}$ )
- $OCRn$  – wartość rejestru definiującego rozdzielczość



# Tryby pracy timera0

- Fast PWM – szybki (i mniej dokładny od pozostałych) tryb generowania fali o zmiennym wypełnieniu. Licznik liczy do wartości OCR0 i zostaje wyzerowany. W tym momencie następuje zmiana poziomu logicznego sygnału na pinie OC0 mikrokontrolera.

$$f_{OCnPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

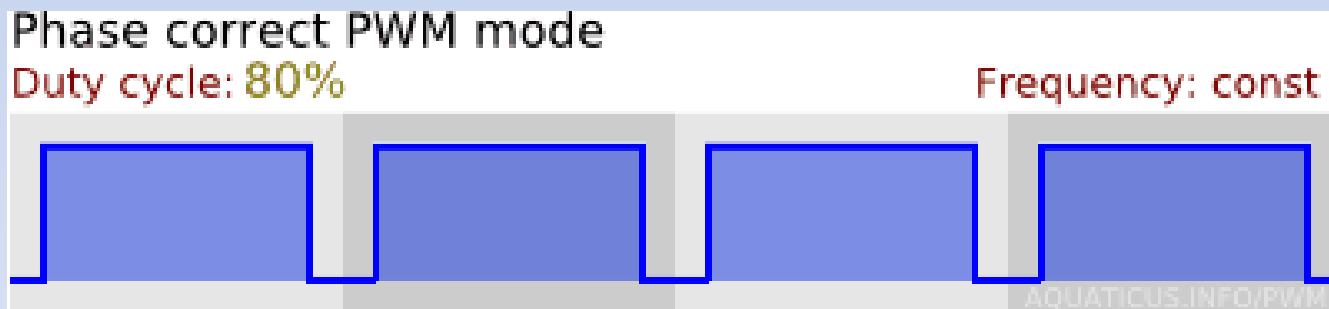


Źródło: [aquaticus.info](http://aquaticus.info)

# Tryby pracy timera0

- Phase Correct PWM – licznik liczy do wartości OCR0 (w górę) i następnie od OCR0 do wartości 0 (w dół). Z tego powodu maksymalna częstotliwość jest dwa razy mniejsza niż w trybie Fast PWM.

$$f_{OCnPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510}$$



Źródło: [aquaticus.info](http://aquaticus.info)

# Dodatkowe możliwości timera1

- Zmienna rozdzielczość (do 16 bitów)
- Dwa wyprowadzenia OC (OC1A i OC1B) – możliwość generowania dwóch sygnałów PWM
- Tryb Phase & Frequency Correct

# Programowy PWM

```
ISR(TIMER1_COMPA_vect) //procedura obsługi przerwania
{
    if(PWM_counter<PWM_value) //jezeli licznik nie doliczył jeszcze do
//ustalonego PWM_value
    {
        PORTB=0xFF; //zapalenie diod led
    }
    else //gdy przekroczono prog rownosc PWM_counter i PWM_value
    {
        PORTB=0x00; //wylaczenie diod led
    }
    PWM_counter++; //zliczanie licznika w gore
}
```