Co-funded by the
Erasmus+ Programme
of the European Union

"Teaching online electronics, microcontrollers and programming in Higher Education"
Erasmus+ Programme
2020-1-PL01-KA226-HE-095653

Podstawy systemów mikroprocesorowych: Przerwania i liczniki

dr inż. Dariusz Tefelski

dariusz.tefelski@pw.edu.pl

Pokój 225GF

Przerwania

- Warunek lub zdarzenie, które przerywa normalny ciąg instrukcji w programie.
- Przerwanie wywołuje między instrukcjami skok do procedury obsługi przerwania.
- Gdy procedura obsługi przerwania kończy się, wznawiany jest normalny ciąg instrukcji w programie

Źródła przerwań

- Przerwania dzielimy w ogólności na:
  - wewnętrzne lub zewnętrzne,
  - programowe lub sprzętowe.
- Przerwanie zewnętrzne jest wywoływane przez urządzenie/układ poza mikrokontrolerem.
- Przerwanie wewnętrzne jest wywoływane przez jeden z podukładów mikrokontrolera.
- Przerwanie sprzętowe zachodzi w wyniku zmiany stanu jednego z układów fizycznych.
- Przerwanie programowe zachodzi w wyniku wykonania instrukcji programu.

System przerwań w mikrokontrolerze AVR

- ATMega32 reaguje na 21 różnych rodzajów przerwań.
- Przerwania są numerowane zgodnie z ich priorytetem od 1 do 21.
- Przerwanie RESET ma numer 1.
- Każde przerwanie powoduje skok do specyficznego dla niego adresu w pamięci programu.
  - Procedura przerwania RESET ma adres $0000.

Wektory przerwań

- Procedura przerwania k jest umieszczona pod adresem $2(k-1)$ w pamięci programu:
  - adres \$0000 - przerwanie RESET,
  - adres \$0002 - przerwanie zewnętrzne 0,
  - Adres \$0004 - przerwanie zewnętrzne 1.
- Ponieważ jest miejsce tylko na jedną lub dwie instrukcje, procedura przerwania rozpoczyna się od skoku w inne miejsce pamięci programu, gdzie znajduje się reszta kodu.

Table 18. Reset and Interrupt Vectors

Vector No.	Program Address[2]	Source	Interrupt Definition
1	$000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$002	INT0	External Interrupt Request 0
3	$004	INT1	External Interrupt Request 1
4	$006	INT2	External Interrupt Request 2
5	$008	TIMER2 COMP	Timer/Counter2 Compare Match
6	$00A	TIMER2 OVF	Timer/Counter2 Overflow
7	$00C	TIMER1 CAPT	Timer/Counter1 Capture Event
8	$00E	TIMER1 COMPA	Timer/Counter1 Compare Match A
9	$010	TIMER1 COMPB	Timer/Counter1 Compare Match B
10	$012	TIMER1 OVF	Timer/Counter1 Overflow
11	$014	TIMER0 COMP	Timer/Counter0 Compare Match
12	$016	TIMER0 OVF	Timer/Counter0 Overflow
13	$018	SPI, STC	Serial Transfer Complete
14	$01A	USART, RXC	USART, Rx Complete
15	$01C	USART, UDRE	USART Data Register Empty
16	$01E	USART, TXC	USART, Tx Complete
17	$020	ADC	ADC Conversion Complete
18	$022	EE_RDY	EEPROM Ready
19	$024	ANA_COMP	Analog Comparator
20	$026	TWI	Two-wire Serial Interface
21	$028	SPM_RDY	Store Program Memory Ready

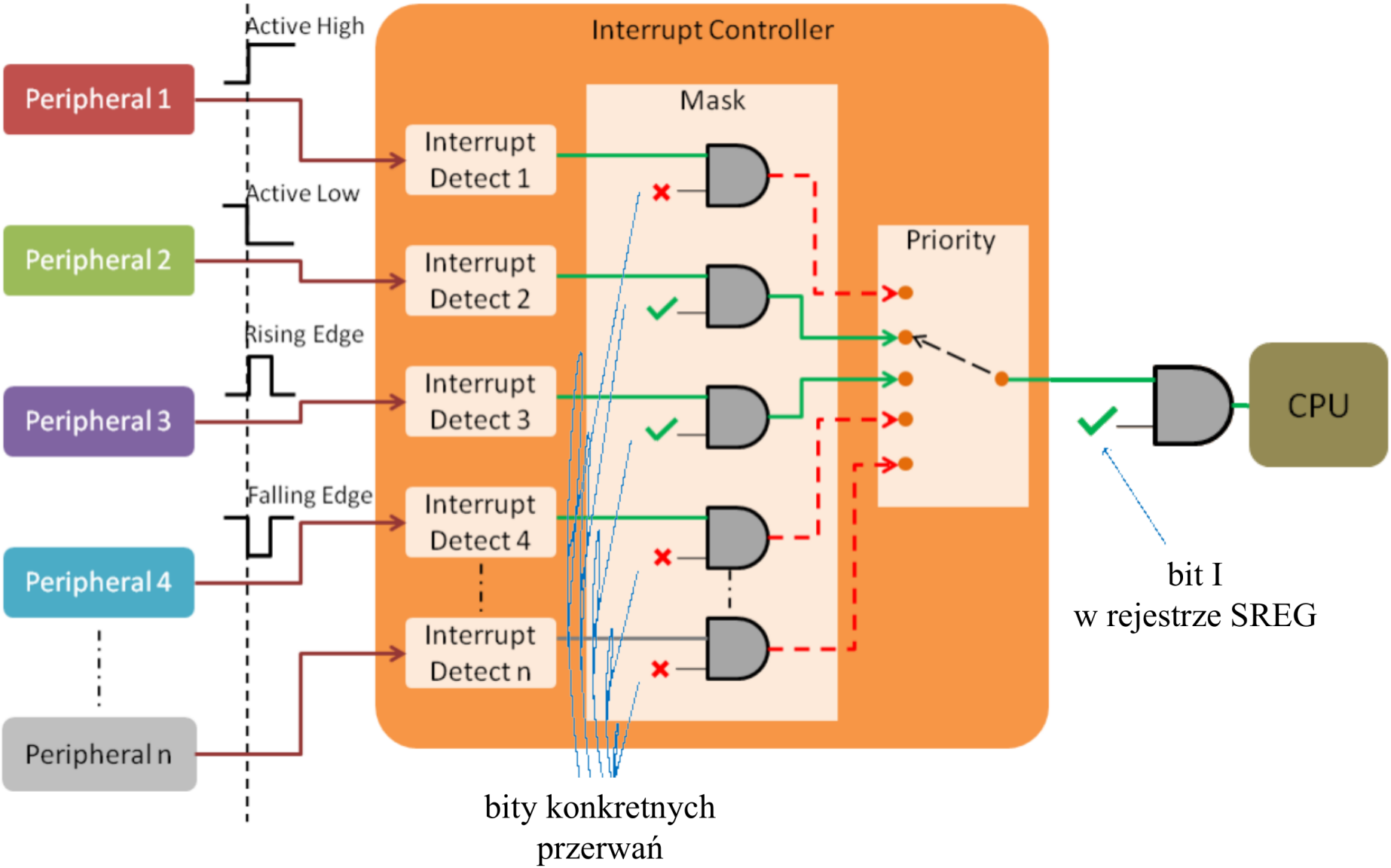
Włączanie przerwań

- Każde potencjalne źródło przerwań można indywidualnie włączać i wyłączać.
- Wyjątek to przerwanie RESET, które nie można wyłączyć.
- By przerwania mogły działać, trzeba ustawić globalny bit przerwań w rejestrze SREG.
- Przerwanie RESET zachodzi niezależnie od ustawienia tego bitu.

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/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	

Działanie przerwań

- Jeżeli:
  - globalny bit przerwań jest ustawiony,
  - **oraz** bit konkretnego przerwania jest ustawiony,
  - **oraz** zaszedł warunek przerwania $\rightarrow$ **to wystąpi przerwanie.**
- Co się właściwie dzieje po zakończeniu aktualnej instrukcji?
  - Aktualna wartość licznika programu PC odkładana jest na stos.
  - Kasowany jest globalny bit przerwań.
  - Właściwy adres procedury obsługi przerwania jest umieszczany w liczniku programu.

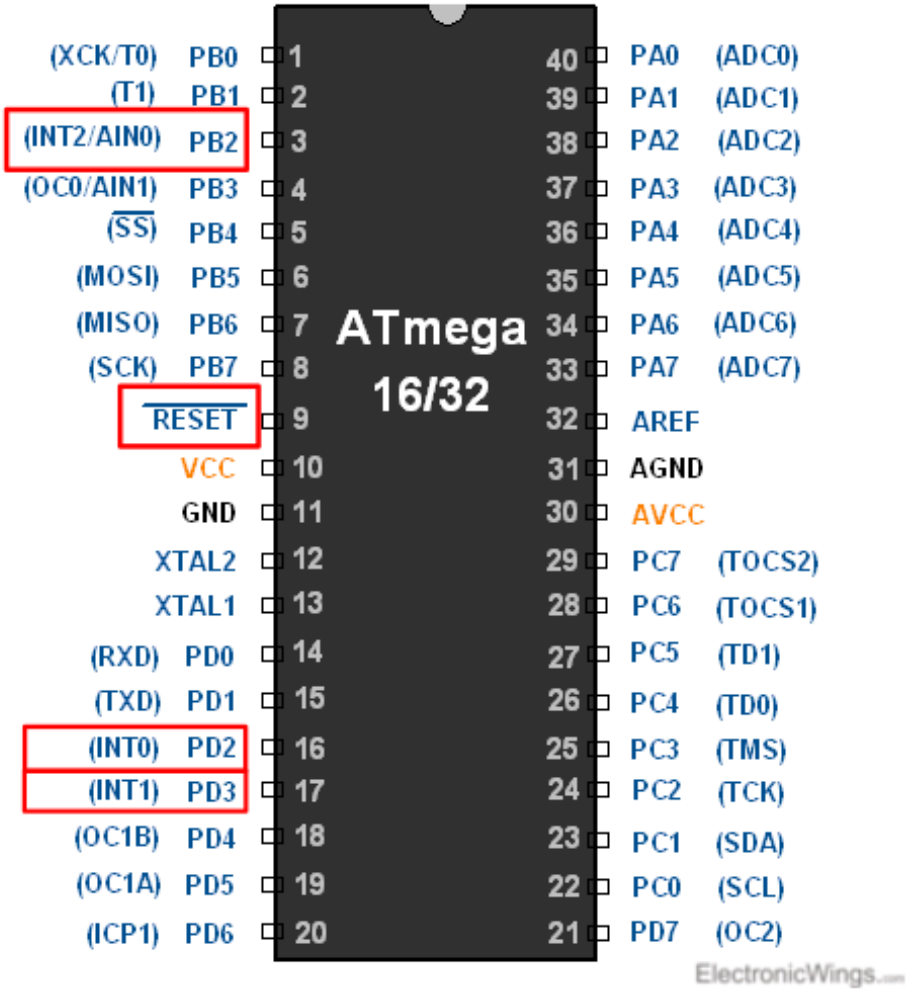

Active High
Peripheral 1
Active Low
Peripheral 2
Rising Edge
Peripheral 3
Falling Edge
Peripheral 4
Peripheral n
Interrupt Controller
Mask
Interrupt Detect 1
Interrupt Detect 2
Interrupt Detect 3
Interrupt Detect 4
Interrupt Detect n
Priority
CPU
bit I
w rejestrze SREG
bity konkretnych
przerwań

Powrót z procedury przerwania

- Po zakończeniu procedury obsługi przerwania:
  - adres ze stosu jest umieszczany w liczniku programu,
  - ustawiany jest globalny bit przerwań.
- To powoduje kontynuację uprzednio przerwanego programu:
  - Przynajmniej jedna instrukcja zostanie wykonana zanim nastąpi inne przerwanie.

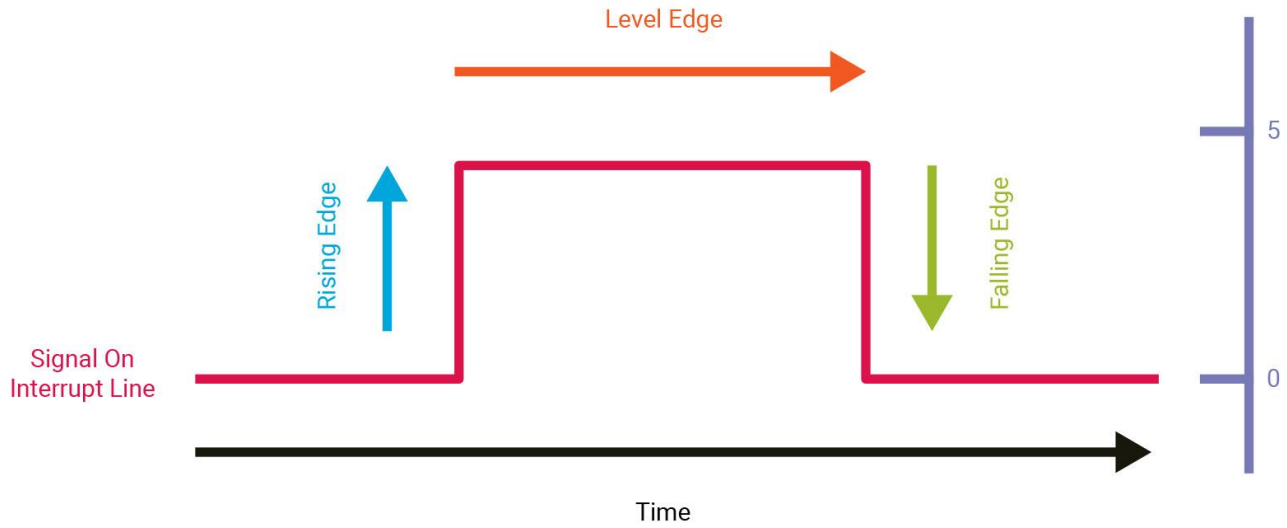
Przerwania zewnętrzne

- ATMega32 reaguje na 4 różne przerwania zewnętrzne - sygnały wprowadzone na odpowiednie wejścia mikrokontrolera:
  1. RESET (pin 9),
  2. INT0 (pin 16 - PD2),
  3. INT1 (pin 17 - PD3),
  4. INT2 (pin 3 - PB3).



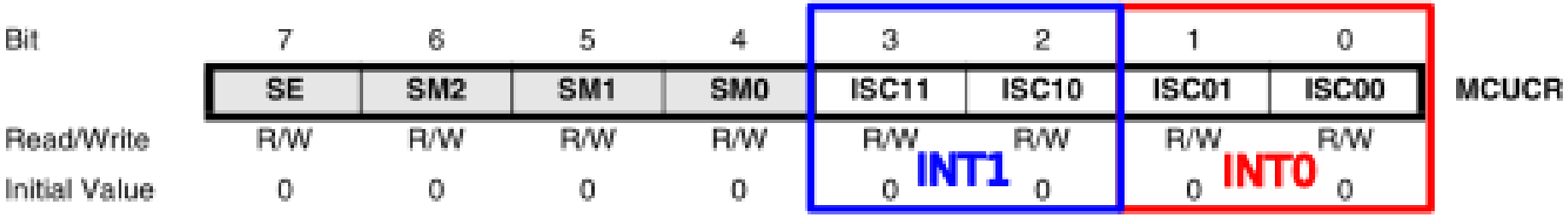
Konfiguracja przerwań zewnętrznych

- Wyzwalane poziomem:
  - gdy stan wejścia jest niski (logiczne 0).
- Wyzwalane zboczem:
  - stan wejścia zmienił się,
  - zboczem opadającym (z 1 do 0),
  - zboczem rosnącym (z 0 do 1).



Source: `https://diyodemag.com/education/interrupts_make_your_code_more_responsive_to_outside_stimulus`

MCUCR (MCU Control Register)



ISC (Interrupt Sense Control bits)

Table 35. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

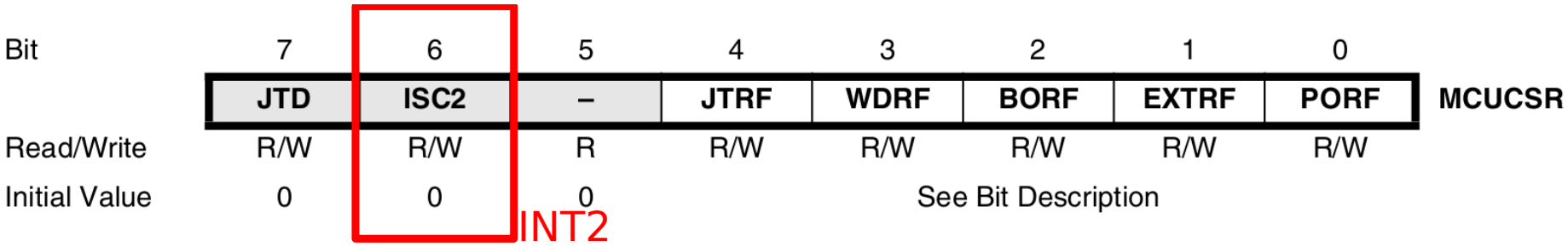
Wyzwalanie poziomem

- Procesor próbkuje poziomy na pinach INT0 i INT1 w każdym cyklu zegara.
- Impulsy krótsze niż jeden cykl mogą być niezauważone.
- Przerwanie niskim poziomem nastąpi tylko, gdy niski stan na wejściu będzie po zakończeniu aktualnie wykonywanej instrukcji.

MCUCSR - rejestr kontroli i statusu mikrokontrolera (Control and Status Register)

Jedyne możliwości wywołania przerwania INT2:

- 0 - zbocze opadające,
- 1 - zbocze rosnące.



GICR - główny rejestr kontroli przerwań (General Interrupt Control Register)

Tu włączamy przerwania zewnętrzne

Bit	7	6	5	4	3	2	1	0	
	INT1	**INT0**	**INT2**	**–**	**–**	**–**	**IVSEL**	**IVCE**	**GICR**
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

GIFR - główny rejestr flag przerwań (General Interrupt Flag Register)

Bit	7	6	5	4	3	2	1	0	
	INTF1	INTF0	INTF2	–	–	–	–	–	GIFR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- Ustawiona flaga (1) oznacza, że wystąpiło żądanie przerwania przez urządzenie zewnętrzne.
- Te flagi używane są tylko dla przerwań zboczem.
- Gdy mikrokontroler przechodzi do procedury obsługi,następuje zerowanie znacznika.
- Gdy nie korzystamy z przerwań, flagi nadal mogą być ustawiane. Choć samo żądanie jest ignorowane, to możemy wykorzystać tę informację w programie (odpytywanie ? pooling). Zerowanie trzeba wykonać manualnie (UWAGA: PRZEZ WPISANIE 1!!!!).

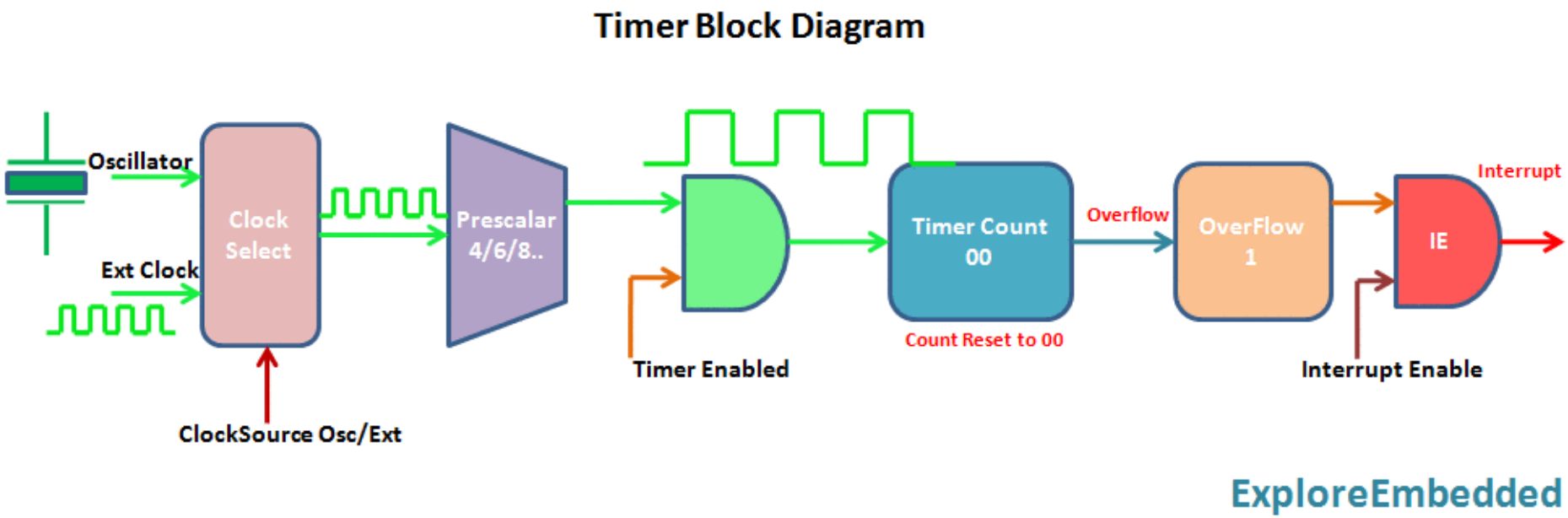
Przerwania programowe

- Jeśli piny przerwań zewnętrznych są skonfigurowane jako wyjścia, możemy wpisać do nich 0 i 1 programowo.
  - To może wyzwolić przerwania zgodnie z ustawieniami dla przerwań zewnętrznych.

Liczniki (timers/counters)

- W mikrokontrolerze ATMega32 dostępne są trzy moduły liczników (timerów), dwa 8-bitowe i jeden 16-bitowy. Oprócz tych istnieje także specjalizowany licznik tzw. *Watchdog Timer*.
- Stan każdego licznika jest przechowywany w odpowiednim rejestrze.
- Licznik może być taktowany zewnętrznym lub wewnętrznym sygnałem zegarowym.
- Licznik może stanowić źródło przerwań (przepełnienie, zrównanie).

- **Jednostka zliczająca:** Główna częścią licznika jest 8-bitowa programowalna dwukierunkowa jednostka logiczna. W zależności od trybu pracy licznik jest zwiększany, zmniejszany lub zerowany w każdym cyklu zegara (clkTn). Bit przepełnienia (TOVn) może być użyty do generowania przerwań.

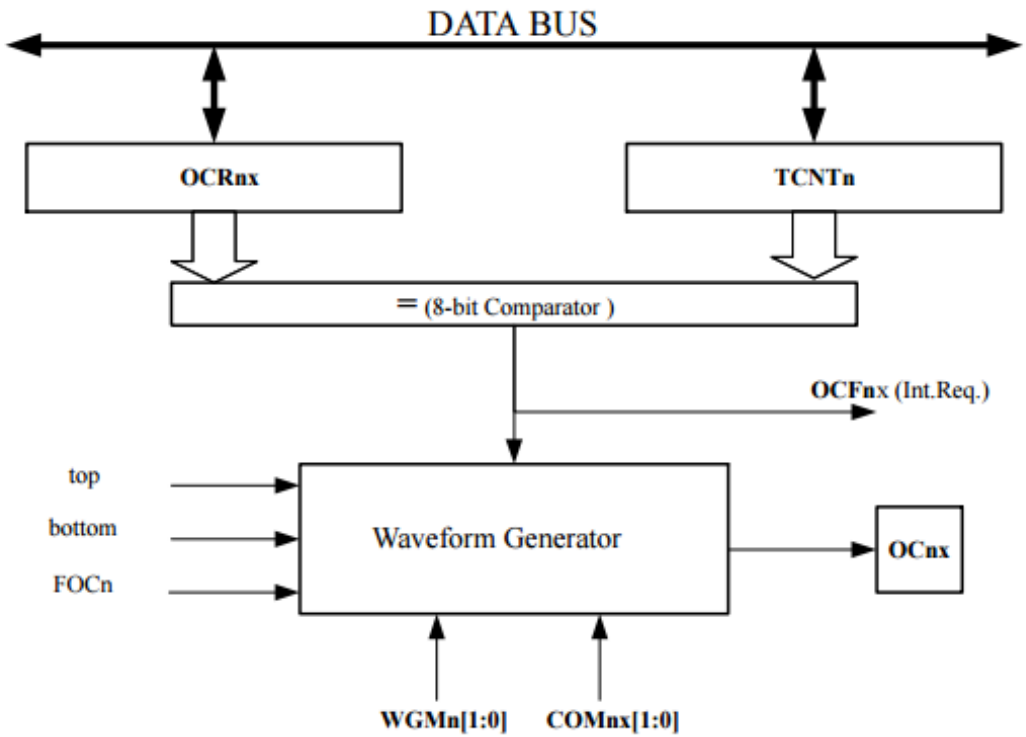


Zdarzenia licznika

- Przekroczenie zakresu (*overflow*):
  - Przekroczenie zakresu występuje, gdy zliczana wartość przekracza 0xFF i staje się równa 0x00.
- Osiągnięcie wartości zadanej (*compare match*):
  - Gdy wartość licznika równa jest zawartości odpowiedniego rejestru specjalnego.

Komparator (compare unit)

Wyjście rejestru komparatora (OCRn) jest stale porównywane z zawartością licznika. Wynik porównania może być użyty przez generator przebiegu do wytwarzania PWM lub sygnału o zmiennej częstotliwości na wyjściu porównania (OCn). Wynik porównania będzie również ustawiał Bit Porównania (OCFn), który może być użyty do wytwarzania przerwania (od porównania).



Źródło: https://microchipdeveloper.com/8avr:avrdblbuf

Odpytywanie liczników (pooling)

- Stan licznika można badać w programie głównym poprzez tzw. *pooling*.
- Odczyt flag w rejestrze TIFR (*Timer Interrupt Flag Register*).
- Przekroczenie zakresu lub osiągnięcie wartości zadanej zmieniają stan odpowiednich bitów w rejestrze TIFR:
  - TOVn oraz OCFn (n=0, 1, lub 2).
  - Licznik 1 (16-bitowy) ma dwa rejestry porównań OCR: 1A oraz 1B.

Automatyczne funkcje liczników

- Liczniki 0 i 2 (8-bitowe) można skonfigurować tak, by automatycznie kasowały, ustawiały, lub odwracały stan odpowiednich wyjść mikrokontrolera, gdy nastąpi przyrównanie do rejestru OCRn.
  - Nie wymaga to czasu obliczeniowego CPU ani obsługi przerwania - wszystko na poziomie sprzętowym.
  - Odpowiedni pin OCnx musi być ustawiony jako wyjście. Normalne działanie tego pinu jest niemożliwe:
    - OC0 (PB3), OC2 (PD7).
    - OC1A (PD5), OC1B (PD4).

Źródła zegarowe liczników

- Liczniki do zliczania mogą wykorzystać zegar wewnętrzny lub zewnętrzny sygnał zegarowy.
- Częstotliwość zegara wewnętrznego może być zredukowana (preskalowana), dzięki czemu liczniki są taktowane wolniej.
  - Możliwy podział częstotliwości przez 8, 64, 256, 1024.
  - Timer2 umożliwia również preskalowanie sygnału zewnętrznego.

Rejestr kontrolny licznika 0 (TCCR0 Timer/Counter 0 Control Register)

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	

Bity 6,3 - WGM01:0 - rodzaje generowanych przebiegów:

- Bity te kontrolują sposób zliczania licznika: wartość maksymalną do której dąży licznik i rodzaj wytwarzanego przebiegu.
- Tryby pracy licznika:
  - zwykły,
  - zerowanie licznika podczas porównania,
  - dwa rodzaje PWM (modulacji szerokości impulsu).

Table 38. Waveform Generation Mode Bit Description(1)

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set-on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Rejestr kontrolny licznika 0 (TCCR0 Timer/Counter 0 Control Register)

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	

Bity 5, 4 - COM01:0 - rodzaj wyjścia dla trybu porównania:

- Określają zachowanie się wyjścia komparatora licznika (końcówka OC0).
- Jeżeli jeden z bitów lub oba bity są ustawione to wyjście układu porównującego w liczniku zostaje dołączone do końcówki portu I/O. Rejestr kierunku portu (DDR) musi ustawić port jako wyjście. Gdy OC0 jest dołączone do końcówek portu, funkcje bitów COM01:0 zależą od trybu pracy licznika (czyli od ustawienia bitów WGM01:0).

Table 39. Compare Output Mode, non-PWM Mode

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Toggle OC0 on compare match
1	0	Clear OC0 on compare match
1	1	Set OC0 on compare match

Rejestr kontrolny licznika 0 (TCCR0 Timer/Counter 0 Control Register)

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	

Bity 2:0 - CS02:0 - wybór sygnału zegarowego:

- Te trzy bity decydują o źródle sygnału zegarowego dla licznika TC0.
- Jeżeli ustawione jest zewnętrzne źródło zegara dla licznika TC0, licznik będzie mógł być napędzany nawet jeżeli port ustawiony jest jako wyjście. Umożliwia to programowe taktowanie licznika (przez zmianę stanu końcówki portu).

Rejestr kontrolny licznika 0 (TCCR0 Timer/Counter 0 Control Register)

Table 42. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Rejestry TCNT0 i OCR0

- W każdej chwili stan licznika można odczytać (albo zapisać) w rejestrze TCNT0.
- Modyfikacja w czasie pracy licznika nie jest wskazana.

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
Read/Write	R/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	

- Rejestr OCR0 przechowuje bajt stale porównywany z wartością rejestru TCNT0.
- Równość obu liczników może spowodować różne zdarzenia:
  - Przerwanie (Output compare) lub po prostu ustawienie flagi przerwania.
  - Generację przebiegu na nóżce OC0.
  - Restart licznika.

Bit	7	6	5	4	3	2	1	0	
	OCR0[7:0]								OCR0
Read/Write	R/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	

Przerwania licznika

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/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	

Licznik 0:

- OCIE0 - *Output Compare Interrupt Enable 0*,
- TOIE0 - *Timer Overflow Interrupt Enable 0*.

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/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	

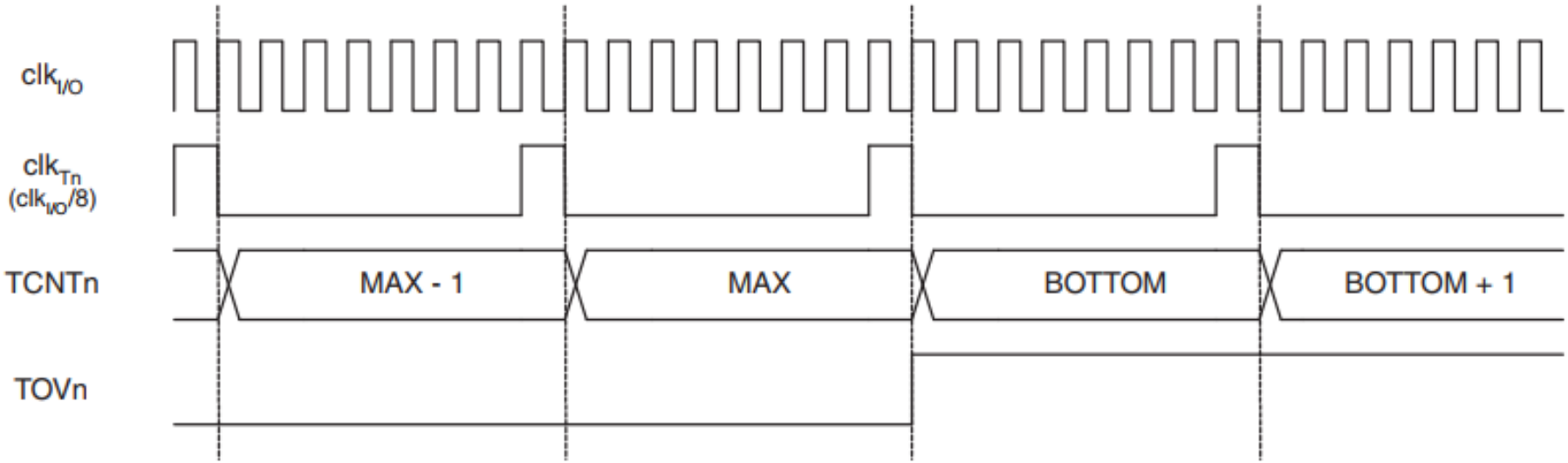
Licznik 0:

- OCF0 - *Output Compare Flag*,
- TOV0 - *Timer Overflow Flag*.

Tryby pracy licznika

- Tryb normalny:
  - licznik liczy do góry,
  - flaga przerwania ustawiona, gdy licznik osiągnie 0.
- Tryb porównania (CTC):
  - licznik liczy do góry, aż osiągnie wartość rejestru OCR0,
  - w następnym kroku licznik przyjmuje wartość 0 i ustawiana jest flaga przerwania.

Tryb normalny (preskaler 8)



Source: https://www.arnabkumardas.com/arduino-tutorial/timer-0-counter-0-concept/

Generacja fali prostokątnej (T=512μs)

```
#include<avr/io.h>
#define F_CPU 8000000

void delay()
{
    TCCR0=(1<<CS01); // Timer Clock = CLK/8 (starts timer too)
    while(!(TIFR&(1<<TOV0))); // Wait Until Overflow
    TIFR=TIFR|(1<<TOV0); // Clear TOV0
    TCCR0=0x00; // Stop Timer0
}

int main(void)
{
    DDRA=0xFF;
    PORTA=0x00;
    TCCR0=0x00;
    TCNT0=0x00;
    while(1){
        PORTA|=(1<<PA0);
        delay();
        PORTA&=~(1<<PA0);
        delay();
    }
}
```

Generacja fali prostokątnej (T=400μs)

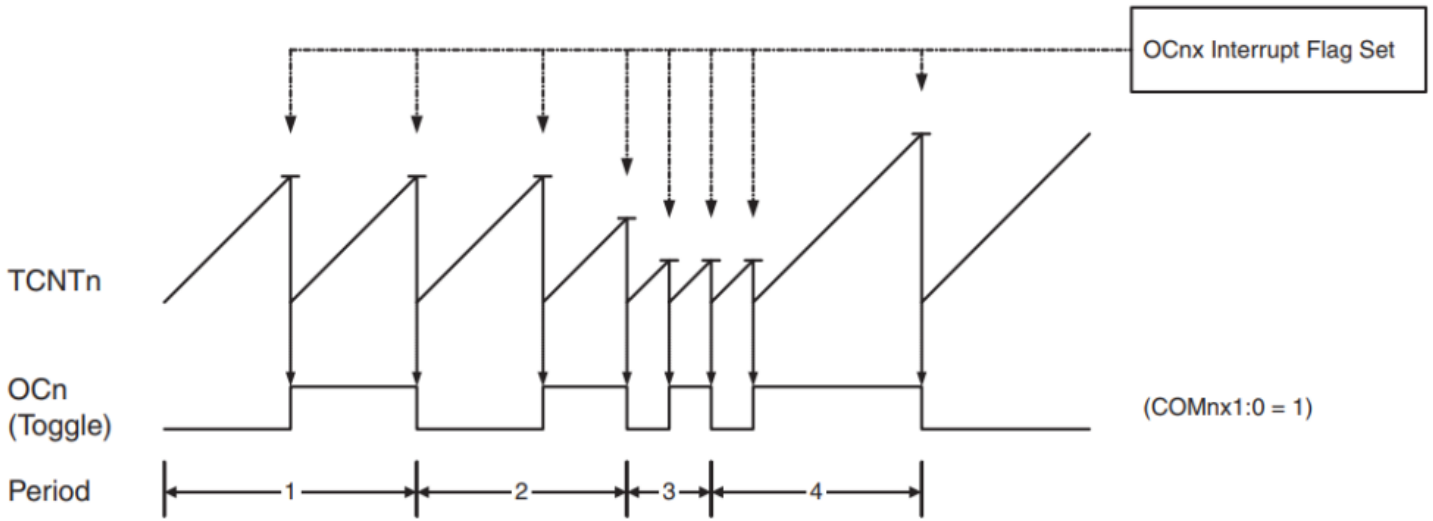
```
void delay()
{
    TCNT0=0x38; //TCNT0=56
    TCCR0=(1<<CS01);
    while(!(TIFR&(1<<TOV0)));
    TIFR=TIFR|(1<<TOV0);
    TCCR0=0x00;
}
```

Tryb CTC (Clear Timer on Compare Match)

- W trybie tym (ustawienie bitów **WGM01:0 = 2**) jest używany rejestr **OCR0**.
- W tym trybie licznik jest zerowany, gdy zliczy do takiej wartości, jaka jest wpisana do rejestru **OCR0**.
- Rejestr ten definiuje największą wartość, do której może doliczyć licznik i w ten sposób decyduje o jego rozdzielczości.

Generowanie przebiegu prostokątnego w trybie CTC

- Przerwanie może być generowane podczas przepełniania się licznika przez użycie flagi **OCF0**.
- Przerwanie może zmienić wartość szczytów dla licznika.
- Ostrożnie ze zmianą wartości szczytowej zbliżonej do zera, podczas gdy licznik pracuje z wyłączonym przeskalerem. Licznik będzie musiał zliczyć do wartości maksymalnej (0xFF) i po wyzerowaniu zliczy do wartości, przy której wystąpi porównanie.
- Generowanie przebiegu prostokątnego w trybie CTC na wyjściu OC0 uzyskuje się przez ustawienie bitów trybu Wyjścia Porównania **(COM01:0 = 1)**.



Source: `https://www.arnabkumardas.com/arduino-tutorial/timer-0-counter-0-concept/`

Tryb CTC

- Generowany przebieg osiągnie najwyższą częstotliwość równą:

$$f_{OC0} = \frac{f_{CLK\_I/O}}{2} \tag{1}$$

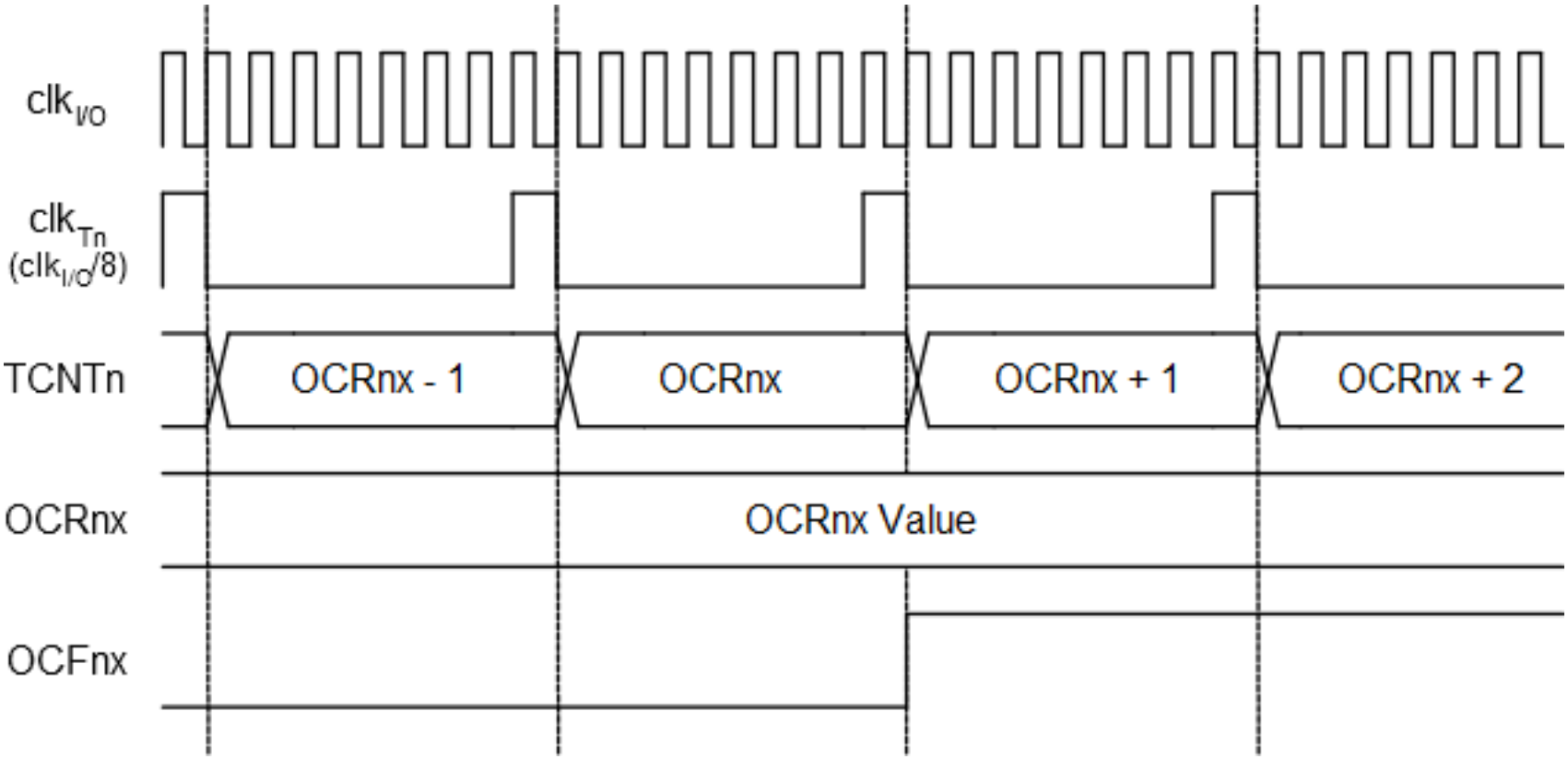
gdzy do **OCR0** wpisane jest zero.

- Częstotliwość przebiegu wyrażona jest wzorem:

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

gdzie: N jest podziałem preskalera (przyjmuje wartości 1, 8, 64, 256 lub 1024).

Tryb CTC (preskaler 8)



Źródło: https://onlinedocs.microchip.com/pr/GUID-173AD72D-41FE-4760-A93C-7078A02BD908-en-US-7/index.html?GUID-082BC7C0-6422-4A11-B254-15BE94219A84

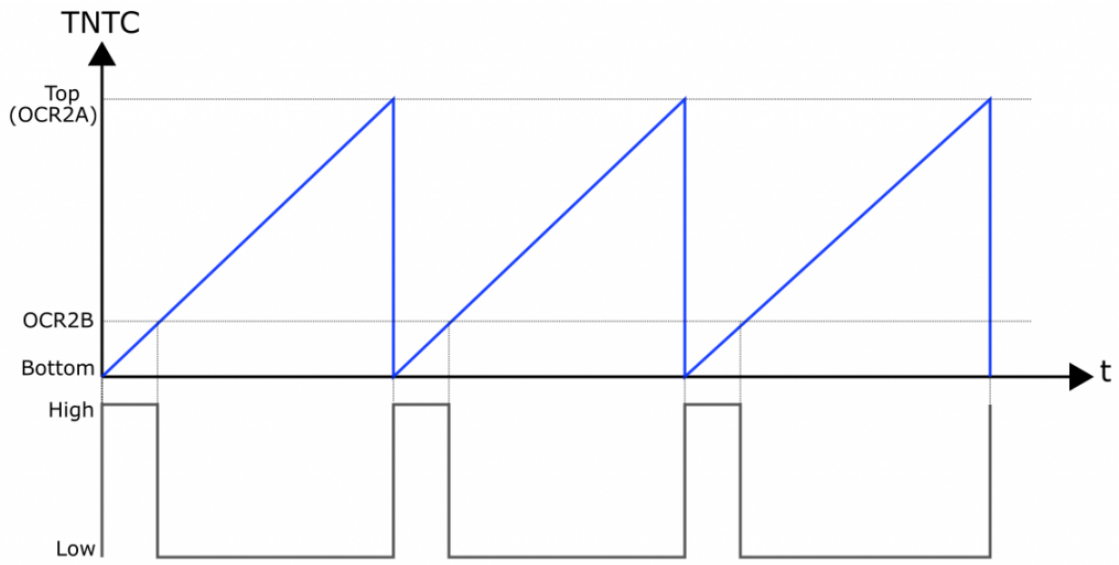
Generacja fali prostokątnej (f=2kHz) w trybie CTC

```
#include <avr/io.h>
#define F_CPU 8000000

int main(void)
{
    DDRB=0xFF;
    PORTB=0x00;
    // Mode: CTC
    TCCR0=(1<<WGM01);
    // CLK/8
    TCCR0|=(1<<CS01);
    // toggle OC0 on compare match
    TCCR0|=(1<<COM00);
    //OCR0=249 f=8000000/(2*8*250)=2KHz
    OCR0=0xF9;
    while (1);
}
```

Szybki tryb Modulacji Szerokości Impulsu (Fast PWM)

- Licznik zlicza od zera do maksymalnej (0xFF) i ponownie zaczyna od zera.
- Wyjście porównania **OC0** jest zerowane, gdy wartość licznika jest większa lub równa wartości w komparatorze i ustawiane gdy licznik jest zerowany.
- Tryb ten jest odpowiedni do regulacji mocy, przetwarzania cyfrowo-analogowego.
- Wysoka częstotliwość PWM umozliwia zmniejszenie rozmiarów zewnętrznych elementów współpracujących - cewek czy kondensatorów i obniża koszty wykonania.



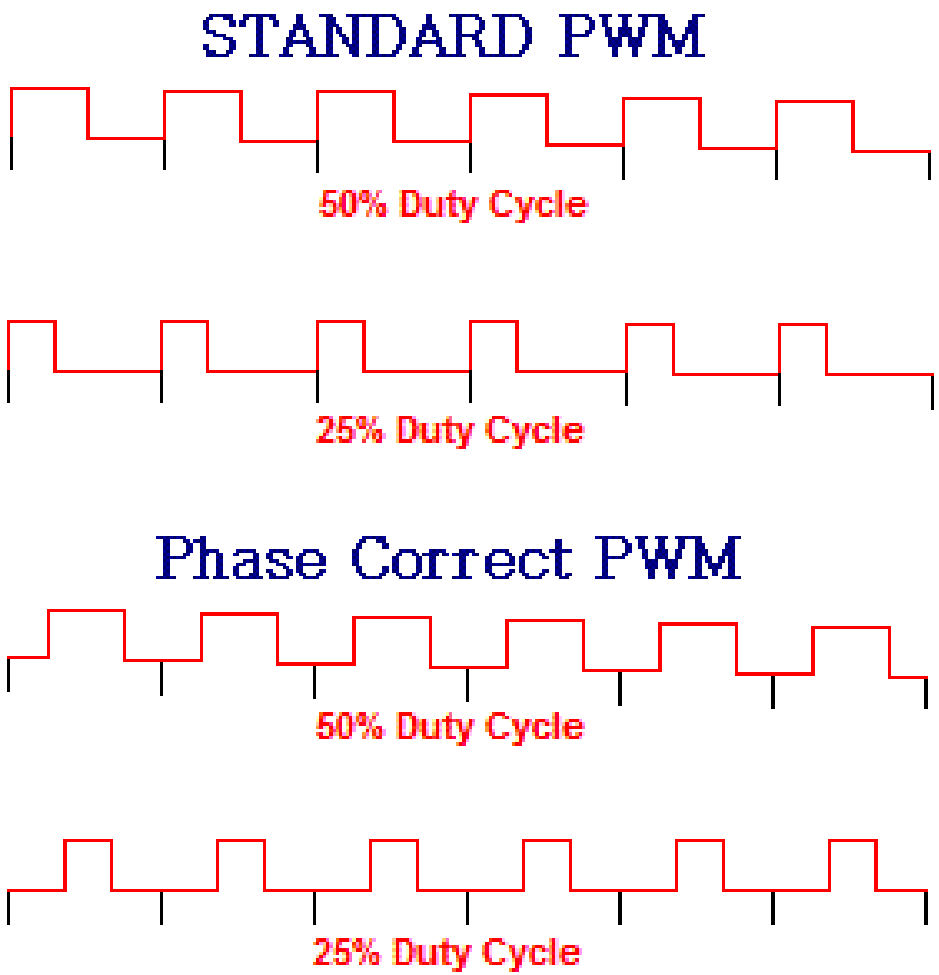
Źródło: `https://wolles-elektronikkiste.de/en/timer-and-pwm-part-1-8-bit-timer0-2`

- Bit przepełnienia **(TOV0)** jest ustawiany podczas osiągnięcia przez licznik wartości maksymalnej. Jeżeli przerwania są aktywne, to procedura przerwania może zmienić wartość rejestru porównania.
- Częstotliwość nośna obliczana jest ze wzoru:

$$f_{OCn} = \frac{f_{CLK\_I/O}}{N \cdot 256} \qquad (3)$$

gdzie: N jest podziałem preskalera (przyjmuje wartości 1, 8, 64, 256 lub 1024).

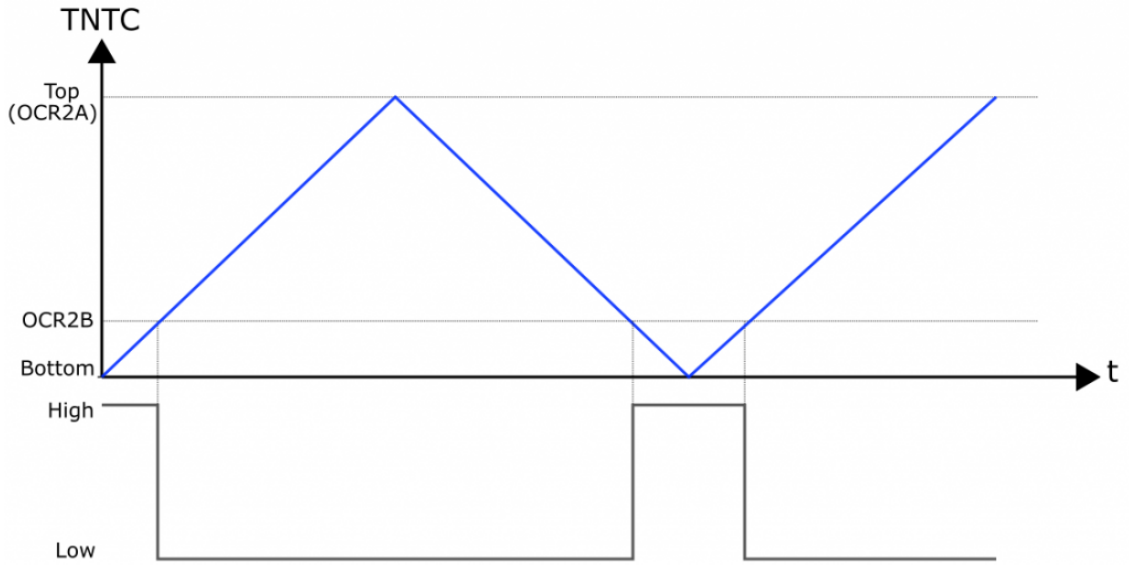
Tryb PWM z Korekcją Fazy



Źródło: `https://www.avrfreaks.net/forum/difference-bewteen-phase-correct-and-fast-pwm`

Tryb PWM z Korekcją Fazy

- Licznik powtarza zliczanie od wartości zerowej do maksymalnej i potem w dół do zera.
- **OC0** jest zerowany przy porównaniu pomiędzy **TCNT0** a **OCR0** podczas zliczania w dół i ustawiany podczas porównania przy zliczaniu w górę.
- System podwójnego zliczania daje mniejszą częstotliwość niż we wcześniejszym trybie.

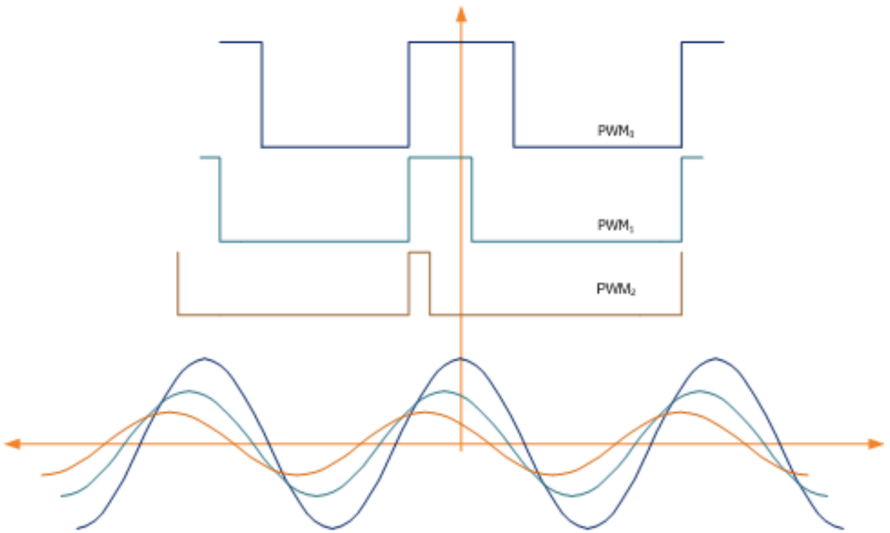


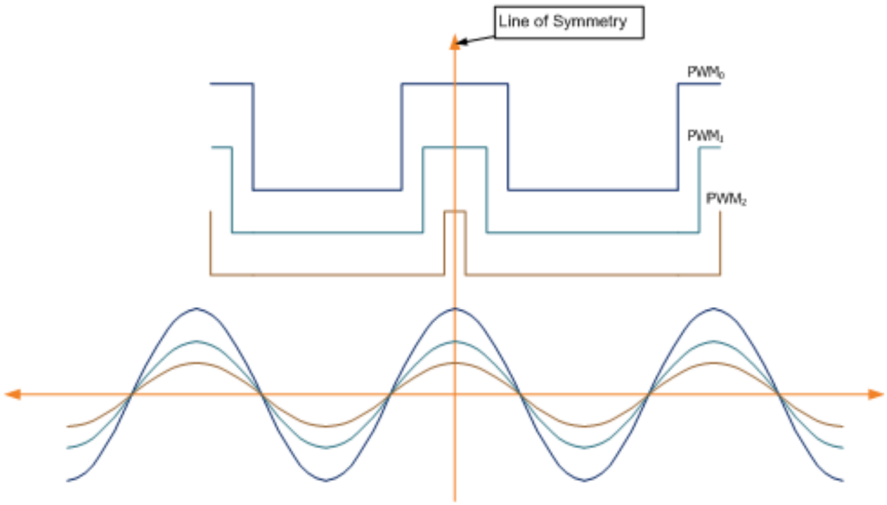
Źródło: `https://wolles-elektronikkiste.de/en/timer-and-pwm-part-1-8-bit-timer0-2`

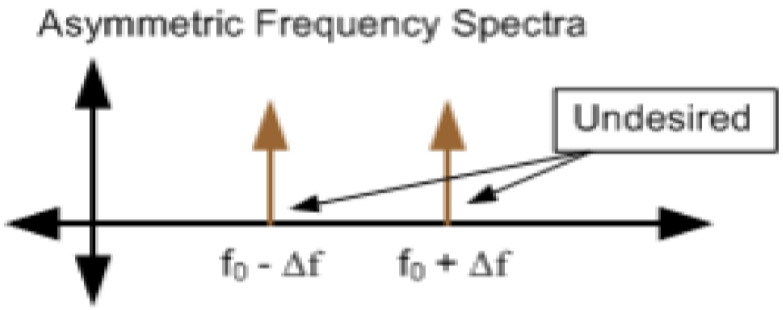
- Dzięki uzyskanej w ten sposób symetrii tryb ten jest zalecany do sterowania silnikami.
- Częstotliwość licznika określona jest z zależności :

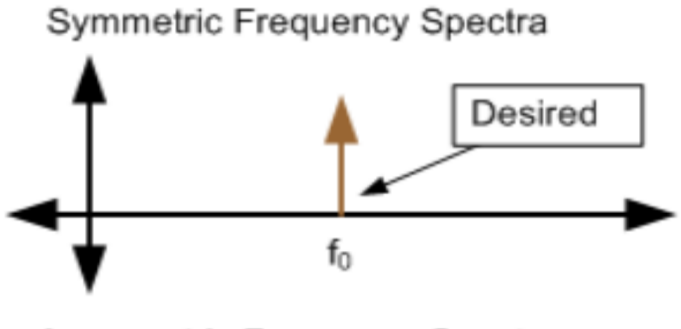
$$f_{OCnPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510} \quad (4)$$

gdzie: N jest podziałem preskalera (przyjmuje wartości 1, 8, 64, 256 lub 1024).


PWM1
PWM0
PWM2


Line of Symmetry
PWM0
PWM1
PWM2


Asymmetric Frequency Spectra
Undesired
f0 - Δf
f0 + Δf

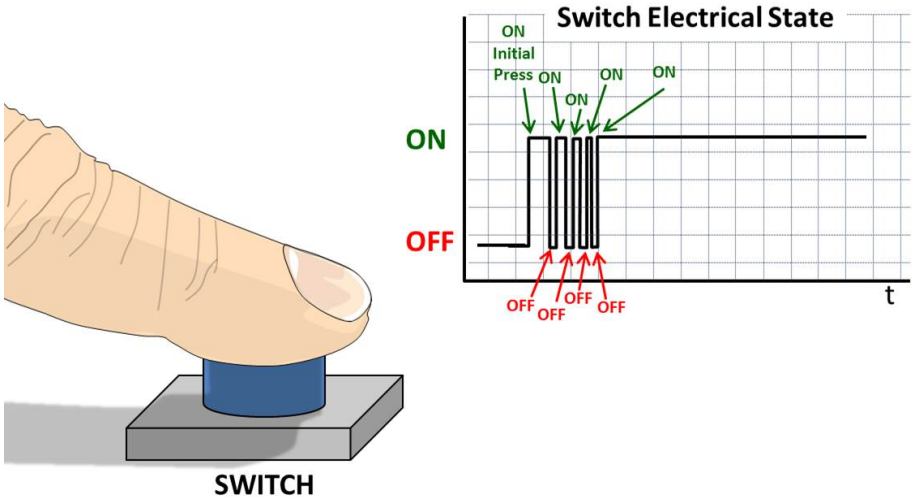

Symmetric Frequency Spectra
Desired
f0

Licznik 1

- licznik 16-bitowy,
- rejestry licznika też 16-bitowe,
- TCNT1H, TCNT1L - wartość licznika (0-65535),
- OCR1AH, OCR1AL - wartość porównania A,
- OCR1BH:OCR1BL - wartość porównania B,
- ICR1H:ICR1L - czas zdarzenia zewnętrznego.

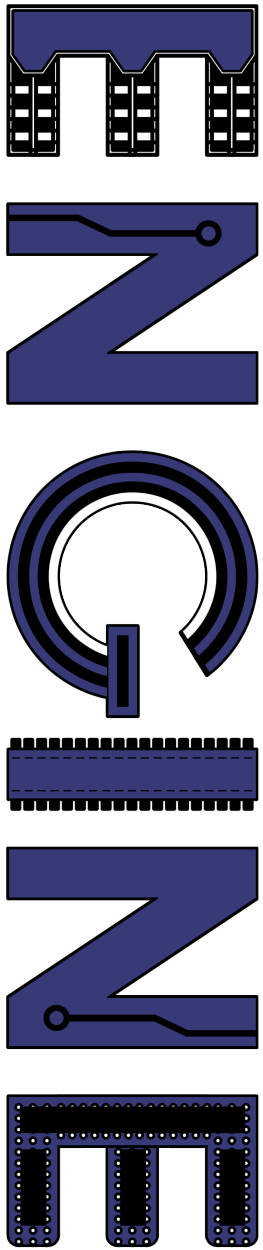
Eliminacja drgań styków

- Naciśnięcie przycisku może wywołać wiele zmian stanów logicznych.
- Efekt ten trwa zwykle nie dłużej niż 10ms.
- W celu eliminacji, należy wykorzystać przerwanie licznika, by odczytywać stan przycisku tylko raz na 10ms (nie używamy funkcji _delay_ms!).
- Stan przycisku zapisać w zmiennej globalnej dostępnej w programie głównym.



Źródło: `https://microchipdeveloper.com/xpress:code-free-switch-debounce-using-tmr2-with-hlt`

Dziękuję za uwagę.



Pierwotna wersja wykładu jest dziełem:
dr hab. Piotra Fronczaka, mgr inż. Marcina Zaręby.