



Zachodniopomorski
Uniwersytet
Technologiczny

WYDZIAŁ ELEKTRYCZNY

Katedra Inżynierii Systemów, Sygnałów i Elektroniki

LABORATORIUM

TECHNIKA MIKROPROCESOROWA

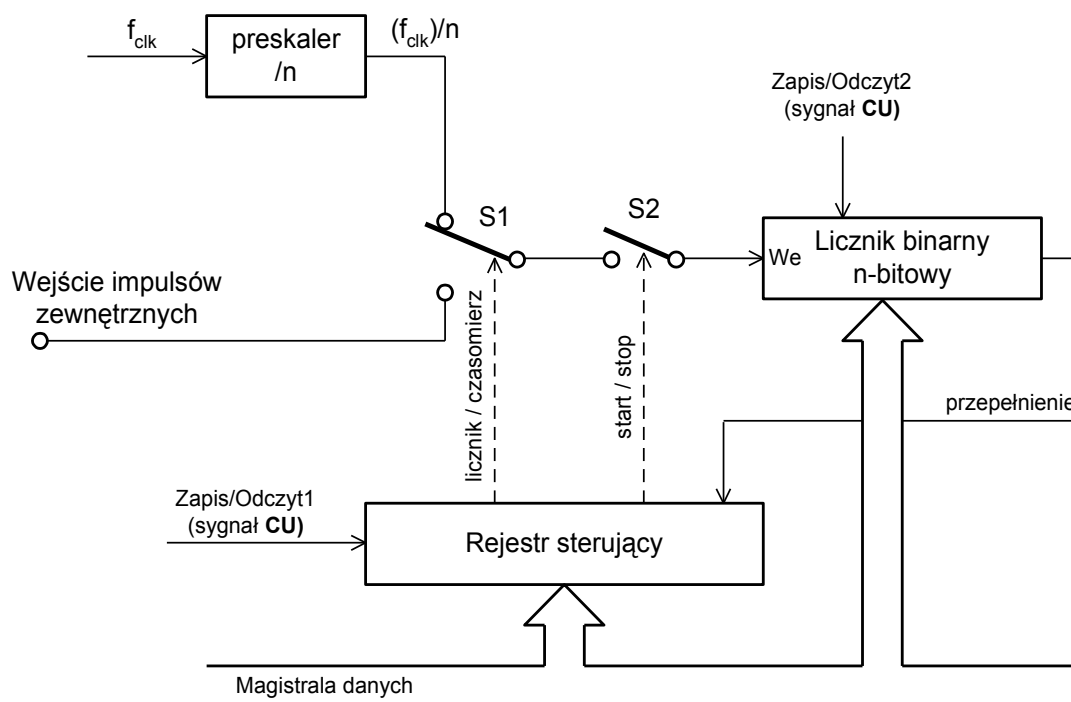
TIMERY

w mikrokontrolerach Atmega16 - 32

Opracował:

mgr inż. Andrzej Biedka

Czasomierz/licznik (ang. Timer/Counter) jest jednym z podstawowych elementów wyposażenia mikrokontrolera. Schemat podstawowej struktury timera przedstawia rysunek 2-1.



Rys. 2-1. Schemat blokowy timera w mikrokontrolerze

Timer składa się z programowanego licznika binarnego oraz współpracującego z nim rejestru sterującego. Zarówno licznik jak i rejestr sterujący są przyłączone do wewnętrznej magistrali danych, dzięki czemu jednostka sterująca mikrokontrolera (CU) może w dowolnym momencie modyfikować lub odczytać ich zawartość.

Licznik binarny najczęściej zlicza w górę – dodaje impulsy podawane na wejście. Impulsy wejściowe są bramkowane (załączane) łącznikiem S2 sterowanym sygnałem **start/stop**, pochodzącym z rejestru sterującego.

Źródłem impulsów zliczanych może być:

- wewnętrzny układ dzielnika częstotliwości sygnału taktującego pracę jednostki centralnej - f_{clk} , tzw. **preskalera**. W takim przypadku, wobec wzorcowej długości czasu impulsu zliczanego pochodzącego najczęściej z generatora stabilizowanego rezonatorem kwarcowym, mówimy o funkcji **czasomierza** lub z języka angielskiego **timera**. Funkcja odpowiada górnemu położeniu przełącznika S1. Preskaler może być dzielnikiem programowanym, wówczas będzie przyłączony do magistrali danych.
- wyprowadzenie mikrokontrolera (ang. **pin**), na który można przyłączyć sygnał pochodzący z zewnątrz. Wówczas urządzenie pełni funkcję **licznika** (ang. **counter**). Funkcja odpowiada dolnemu położeniu przełącznika S1.

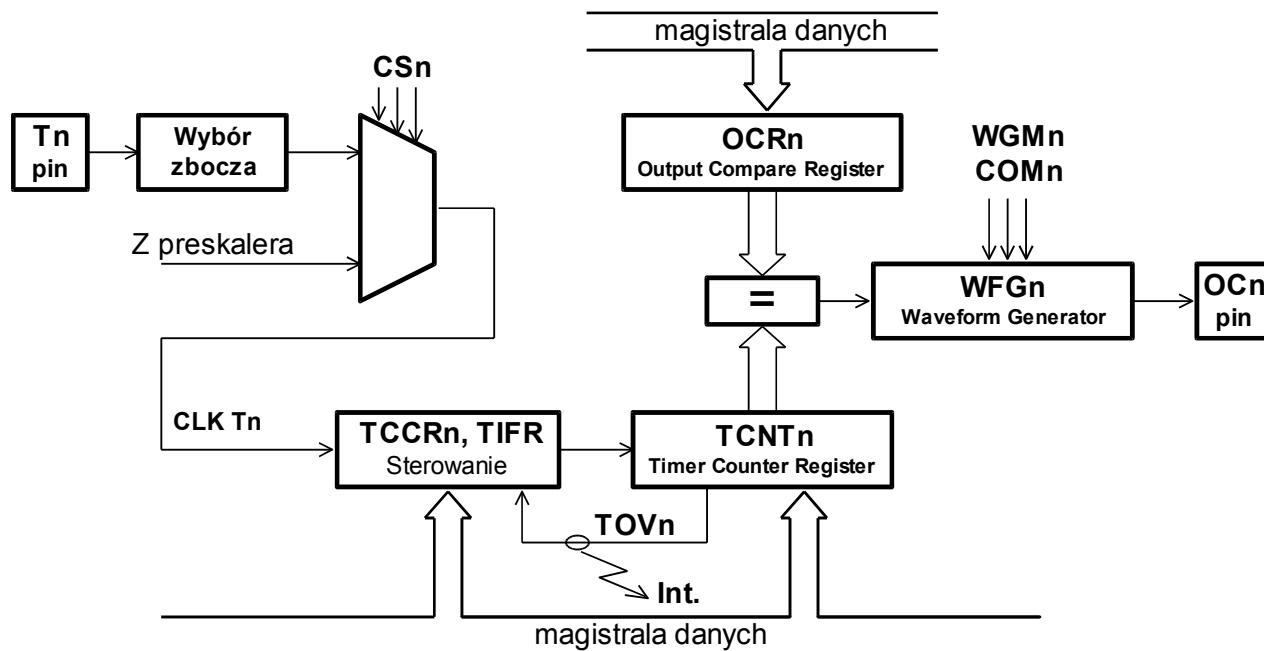
Sygnałem wyjściowym może być stan licznika lub – częściej – jego przepelnienie sygnalizowane zmianą stanu jednego z bitów rejestru sterującego. Dodatkowo przepelnienie licznika binarnego może wywołać stan przerwania.

Należy zaznaczyć, że z uwagi na stosowanie w układach mikrokontrolerów kilku zespołów timerów (mogących pracować niezależnie lub w niektórych konfiguracjach wspólnie), rejestr sterujący jest zbudowany z kilku rejestrów 8-bitowych.

Istotą pracy timera/licznika jest jego autonomiczność, czyli praca niezależna od programu głównego. Niezależność dotyczy zliczania impulsów, gdyż oczywiście sterowanie pracą timera/licznika odbywa się programowo.

Nowoczesne mikrokontrolery zawierają moduły czasomierzy/liczników o bardzo rozbudowanych funkcjach, np.: generatora PWM – często wielokanałowego, generatora taktującego o programowanej częstotliwości, licznika zdarzeń zewnętrznych.

Schemat blokowy timera 8-bitowego (Timer0) wbudowanego do struktury mikrokontrolerów rodziny Atmega przedstawia rysunek 2-2.



Rys. 2-2

Podstawowym rejestrem czasomierza/licznika jest rejestr **TCNTn**, gdzie **n** oznacza numer timera w mikrokontrolerze. Impulsy zliczane w rejestrze mogą pochodzić z zewnątrz mikrokontrolera (dostarczane są przez wyprowadzenie (ang. pin) oznaczone **Tn**) lub z wstępnego dzielnika częstotliwości (ang. prescaler), dzielącego przebieg zegarowy mikrokontrolera przez wybrany stopień podziału. Rejestr zlicza impulsy zwiększając swój stan w zakresie 0 – 255. Kolejny impuls powoduje przepełnienie rejestru **TCNTn** sygnalizowane ustawieniem bitu **TOVn**. Może być ono też źródłem przerwania. Po przepełnieniu stan rejestru równy jest 0.

Licznik/czasomierz0 może pracować w czterech trybach:

Tabela 2-1

Tryb	Opis trybu	WGM01	WGM00	Maksymalna wartość osiągnięta w cyklu pracy timera
0	Normalny	0	0	0xFF
1	PWM, faza poprawna	0	1	0xFF
2	CTC	1	0	OCR0
3	PWM szybki	1	1	0xFF

Wykorzystanie wyjścia **OC0** (ang. Output Compare 0) jest uzależnione od bitów **COM01**, **COM00** znajdujących się w rejestrze **TCCR0**.

Tabela 2-2

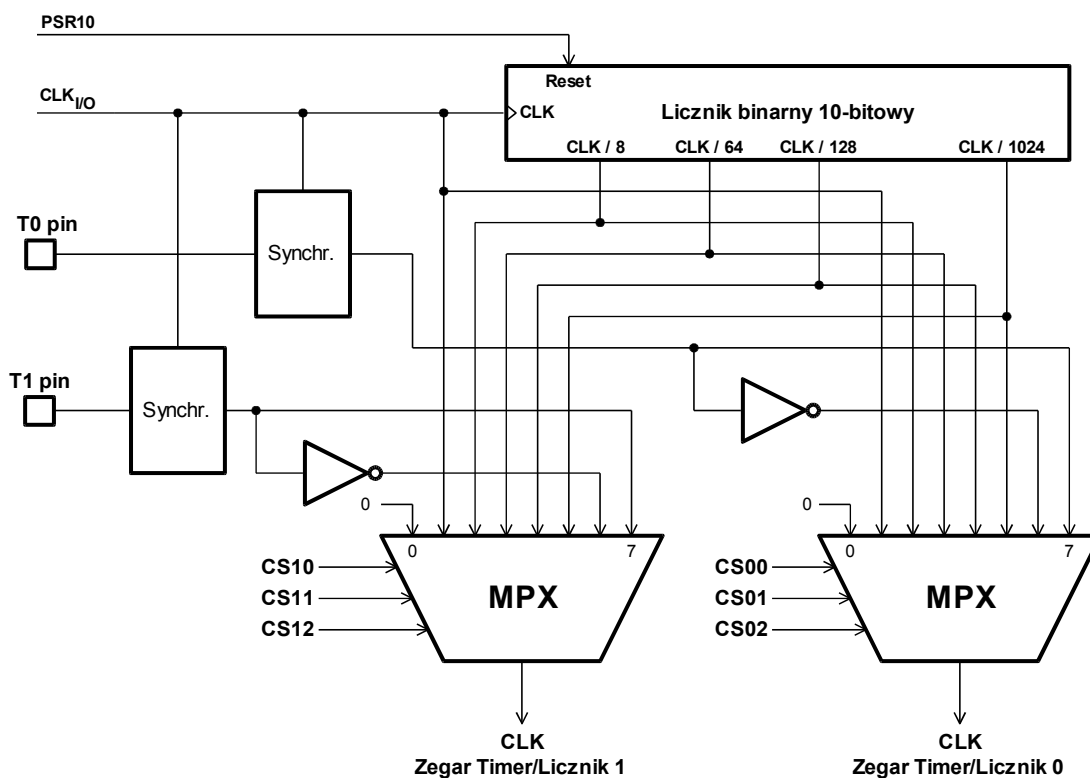
COM01	COM00	Tryb CTC	Tryb Fast PWM
0	0	OC0 wyłączone, funkcja bitu portu	OC0 wyłączone, funkcja bitu portu
0	1	Przełącz OC0 przy zrównaniu TCNT0 z OCR0	Zarezerwowane
1	0	Zeruj OC0 przy zrównaniu TCNT0 z OCR0	Zeruj OC0 przy zrównaniu TCNT0 z OCR0, ustaw OC0 przy TCNT0 = 0
1	1	Ustaw OC0 przy zrównaniu TCNT0 z OCR0	Ustaw OC0 przy zrównaniu TCNT0 z OCR0, zeruj OC0 przy TCNT0 = 0

Rejestr sterujący **TCCR0** zawiera następujące bity:

FOC	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
-----	-------	-------	-------	-------	------	------	------

- FOC** - wymuszenie stanu zgodności porównania TCNT0 i OCR0
- WGM00, WGM01** - wybór trybu pracy czasomierza/licznika. Patrz tabela 2-1
- COM01, COM00** - wybór trybu pracy wyjścia OC0. Patrz tabela 2-2
- CS02-CS00** - wybór stopnia podziału preskalera. Patrz tabela 2-3

Schemat blokowy dzielników wstępnych (preskalerów) czasomierzy 0 i 1 przedstawiony jest na rysunku 2-3.



Rys. 2-3

Ustawienia bitów CS02 – CS00 definiujące stopień podziału preskalerów przedstawia tabela 2-3.

Tabela 2-3

CS02	CS01	CS00	Opis
0	0	0	Zegar zablokowany
0	0	1	$CLK_{I/O}$
0	1	0	$CLK_{I/O} / 8$
0	1	1	$CLK_{I/O} / 64$
1	0	0	$CLK_{I/O} / 256$
1	0	1	$CLK_{I/O} / 1024$
1	1	0	Zegar zewnętrzny T0, aktywne zbocze opadające
1	1	1	Zegar zewnętrzny T0, aktywne zbocze narastające

$CLK_{I/O}$ - zegar taktujący układy wejścia/wyjścia o częstotliwości zgodnej z taktowaniem uC

Rejestr flag **TIFR** zawiera następujące flagi trzech timerów:

OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
-------------	-------------	-------------	--------------	--------------	-------------	-------------	-------------

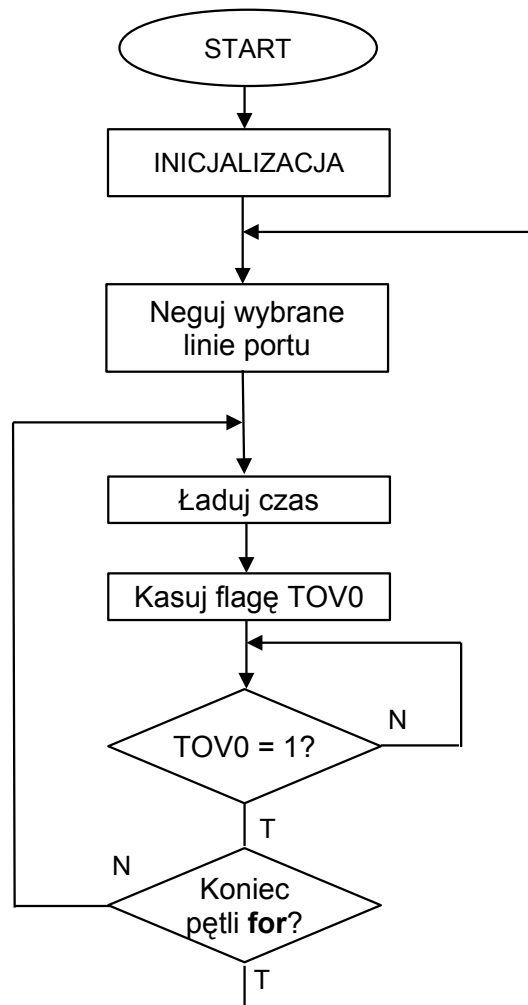
- OCF0** - flaga zgodności porównania TCNT0 i OCR0. Flaga jest kasowana przez wpisanie logicznej jedynki na jej pozycję. Przy przyjęciu przerwania flaga jest kasowana automatycznie.
- TOV0** - flaga przepełnienia timera 0. Flaga jest kasowana przez wpisanie logicznej jedynki na jej pozycję. Przy przyjęciu przerwania flaga jest kasowana automatycznie.

ZADANIA:

1. Na podstawie literatury zapoznać się z pracą czasomierza/licznika0 w trybach Normal, CTC i Fast PWM.
2. Ustalić dla każdego trybu pracy ustawienia rejestrów TCCR0, TCNT0 i OCR0 dla przykładowych wartości czasów odmierzanych przez czasomierz.
3. Zapoznać się z budową, działaniem i programowaniem pracy czasomierza/licznika1 mikrokontrolera ATmega 32.

Przykładowy program wykorzystujący timer będzie generatorem impulsów prostokątnych z obserwacją przebiegu na sygnalizatorze z dowolnymi diodami LED przyłączonymi do portu **PORTB**. Przyjmujemy częstotliwość przebiegu równą 1 Hz. Wykorzystany zostanie timer 0 w trybie Normal.

Algorytm programu przedstawia poniższy rysunek:



pętla **for**: pętla krotności przepełnień timera

Rys. 2-4. Algorytm programu generatora przebiegu prostokątnego

Listing programu napisanego na podstawie algorytmu przedstawiono poniżej.

/* Listing 2-1

Program generatora impulsów prostokątnych o częstotliwości 1 Hz.

Program wykorzystuje Timer0 pracujący w trybie Normal

Zegar uC = 16,0 MHz, preskaler = 1024

Częstotliwość przebiegu taktującego timer: $16,0 \text{ MHz} / 1024 = 15,625 \text{ kHz}$; okres = 64,0 us

Maksymalny czas zliczany przez timer0 = $256 * 64,0 = 16,384 \text{ ms}$

Minimalna krotność przepelnienia timera: $500\text{ms} / 16,384 = 30,52$, przyjmujemy 31

Dla uzyskania okresu 1 sek. przyjmujemy czas timera $500 \text{ ms} / 31 = 16,129\text{ms}$.

Zatem timer musi zliczać: $16,129 \text{ ms} / 64,0 \text{ us} = 252$ takty

*/

```
#include <avr/io.h>
```

```
#define PortLED PORTB
```

```
#define CzasTim0 256 - 252
```

```
int main(void)
```

```
{
```

```
  DDRB = 0xFF; // PORTB ustaw na wyjście
```

```
  TCCR0 = (1 << CS02) | (1 << CS00); // preskaler = 1024,
```

```
  while(1) // pętla główna
```

```
  {
```

```
    PortLED ^= 3; // neguj wybrane bity
```

```
    uint8_t i;
```

```
    for(i = 0; i < 31; i++) // pętla zliczania 31 * 16,129 ms
```

```
    {
```

```
      TIFR |= (1 << TOV0); // kasuj flagę
```

```
      TCNT0 = CzasTim0; // ustaw czas
```

```
      while(~TIFR & (1 << TOV0)); // czekaj na ustawienie flagi
```

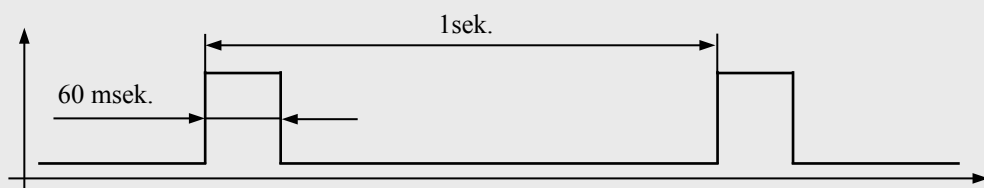
```
    }
```

```
  }
```

```
}
```

ZADANIA:

4. Zmodyfikować program z listingu 2-1 wykorzystując tryb CTC timera0.



Rys. 2-5

5. Napisać program generatora przebiegu prostokątnego o parametrach przedstawionych na rysunku 2-5 z wykorzystaniem timera 0 pracującego w trybie CTC.
6. Napisać program generatora przebiegu prostokątnego o parametrach przedstawionych na rysunku 2-5 z wykorzystaniem timera 0 i timera 1.

Literatura:

- [1] Francuz T. Język C dla mikrokontrolerów AVR. Od podstaw do zaawansowanych aplikacji. Helion, Gliwice, 2011
- [2] Baranowski R. Mikrokontrolery AVR ATmega w praktyce. BTC, Warszawa, 2005
- [3] Kardaś M. Mikrokontrolery AVR. Język C. Podstawy programowania. Atmel, Szczecin, 2011.
- [4] Witkowski A. Mikrokontrolery AVR. Programowanie w języku C. Przykłady zastosowań, PKJS, Katowice 2006
- [5] Karta katalogowa mikrokontrolera Atmega32 firmy ATMEL.