Politechnika Poznańska

Instytut Automatyki i Inżynierii Informatycznej

Zakład Sterowania i Elektroniki Przemysłowej

Programowalne układy licznikowe mikrokontrolera

Programowanie Mikroprocesorów

Materiały do zajęć laboratoryjnych

dr inż. Dominik Łuczak

Dominik.Luczak@put.poznan.pl

I. CEL

WIEDZY

Celem zajęć jest zapoznanie z:

- budową i działaniem programowalnych układów licznikowych mikrokontrolera,
- metodami zliczania impulsów,
- metodami odmierzania czasu,
- systemem obsługi przerwań od układów licznikowych mikrokontrolera.

UMIEJĘTNOŚCI

Celem zajęć jest nabycie umiejętności programistycznych pozwalających na programową obsługę układów licznikowych mikrokontrolera:

- konfiguracja programowalnych układów licznikowych z wykorzystaniem makroinstrukcji,
- zliczanie impulsów zewnętrznych,
- odmierzanie krótkiego czasu,
- wykorzystanie przerwania od układów licznikowych mikrokontrolera,
- odmierzanie długiego czasu,
- prosty program wykorzystujący odmierzanie czasu,
- złożony program wykorzystujący odmierzanie czasu.

KOMPETENCJI SPOŁECZNYCH

Celem zajęć jest kształtowanie właściwych postaw programistycznych:

- tworzenie kodu programu w sposób czytelny,
- stosowanie dyrektyw preprocesora oraz makroinstrukcji,
- prawidłowa obsługa programowa przerwań mikrokontrolera,
- stosowanie stosownych komentarzy,
- weryfikacja poprawności działania programu.

II. POLECENIA KOŃCOWE

Dokończ zadania nie zrealizowane w trakcie zajęć. Przygotuj raport z przeprowadzonych zajęć. Jeden raport może być przygotowany przez maksymalnie 2 osoby. W projekcie zamieść:

1. Opis rozwiązania problemu oraz jego implementację.
2. Rysunki lub fotografie ilustrujące poprawne działanie programu.
3. Opis działania własnych programów. Udokumentuj przeprowadzenie testów funkcjonalnych z wykorzystaniem symulatora.
4. Ważne instrukcje programu należy zaopatrzyć stosownym komentarzem. Wszystkie przygotowane funkcje oraz zmienne globalne należy zaopatrzyć komentarzem zgodnym z składnią generatora dokumentacji (*Doxygen*).

Jako załącznik dołącz spakowane kody programów. Dla każdego zadania przygotuj osobny projekt. Raport należy przygotować i przekazać do kolejnego spotkania. Raporty przekazywane później nie będą oceniane.

Na ocenę raportu będą miały wpływ następujące elementy (łącznie 5 punktów):

1. spełnienie wymogów redakcyjnych (1p),

2. wykonanie, udokumentowanie oraz opis wykonanych zadań (2p),
3. zastosowanie prawidłowego warsztatu programistycznego (2p).

III. PRZYGOTOWANIE DO ZAJĘĆ

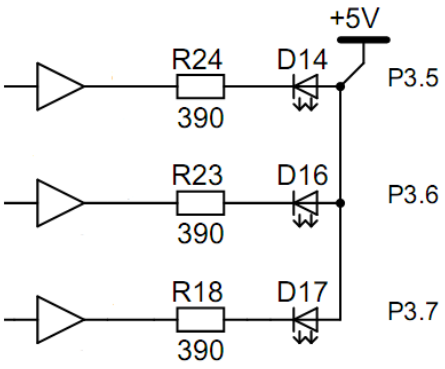
a) ZAPOZNANIE Z PRZEPISAMI BHP

Wszystkie informacje dotyczące instrukcji BHP laboratorium są zamieszczone w sali laboratoryjnej oraz u prowadzącego zajęcia. Wszystkie nieścisłości należy wyjaśnić z prowadzącym laboratorium. Wymagane jest zaznajomienie i zastosowanie do regulaminu.

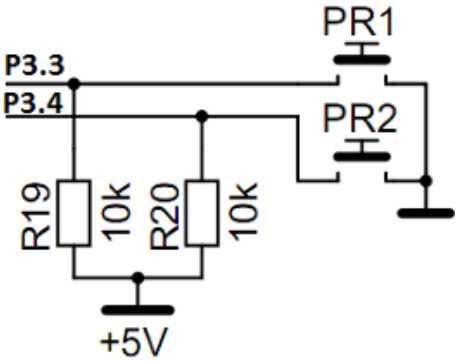
Na zajęcia należy przyjść przygotowanym zgodnie z tematem zajęć. Obowiązuje również materiał ze wszystkich odbytych zajęć.

b) PRZYDATNE SCHEMATY ELEKTRYCZNE

Rys. 1 oraz Rys. 2 przedstawiają fragment schematu elektrycznego zestawu edukacyjnego z mikrokontrolerem ADuC831.



Rys. 1 Schemat połączeń diod



Rys. 2 Podłączenie przycisków monostabilnych do uP

c) WYBRANE REJESTRY MIKROKONTROLERA ADUC831 (TMOD, TCON) [1, 2]

TMOD	Timer/Counter 0 and 1 Mode Register
SFR	Address 89H
Power-On Default Value	00H
Bit Addressable	No

Tab. 1 TMOD SFR Bit Designations

Bit	Name	Function
7	Gate	Timer 1 Gating Control. Set by software to enable timer/counter 1 only while INT1 pin is high and TR1 control bit is set. Cleared by software to enable Timer 1 whenever TR1 control bit is set.
6	C/T	Timer 1 Timer or Counter Select Bit. Set by software to select counter operation (input from T1 pin). Cleared by software to select timer operation (input from internal system clock).
5	M1	Timer 1 Mode Select Bit 1 (Used with M0 Bit).
4	M0	Timer 1 Mode Select Bit 0. M1 M0 0 0 TH1 operates as an 8-bit timer/counter. TL1 serves as 5-bit prescaler. 0 1 16-Bit Timer/Counter. TH1 and TL1 are cascaded; there is no prescaler. 1 0 8-Bit Auto-Reload Timer/Counter. TH1 holds a value which is to be reloaded into TL1 each time it overflows. 1 1 Timer/Counter 1 Stopped.
3	Gate	Timer 0 Gating Control. Set by software to enable timer/counter 0 only while INT0 pin is high and TR0 control bit is set. Cleared by software to enable Timer 0 whenever TR0 control bit is set.
2	C/T	Timer 0 Timer or Counter Select Bit. Set by software to select counter operation (input from T0 pin). Cleared by software to select timer operation (input from internal system clock).
1	M1	Timer 0 Mode Select Bit 1.
0	M0	Timer 0 Mode Select Bit 0. M1 M0 0 0 TH0 operates as an 8-bit timer/counter. TL0 serves as 5-bit prescaler. 0 1 16-Bit Timer/Counter. TH0 and TL0 are cascaded; there is no prescaler. 1 0 8-Bit Auto-Reload Timer/Counter. TH0 holds a value which is to be reloaded into TL0 each time it overflows. 1 1 TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only, controlled by Timer 1 control bits.

TCON Timer/Counter 0 and 1 Control Register
SFR Address 88H
Power-On Default Value 00H
Bit Addressable Yes

Tab. 2 TCON SFR Bit Designations

Bit	Name	Function
7	TF1	Timer 1 Overflow Flag. Set by hardware on a Timer/Counter 1 overflow. Cleared by hardware when the Program Counter (PC) vectors to the interrupt service routine.
6	TR1	Timer 1 Run Control Bit. Set by user to turn on Timer/Counter 1. Cleared by user to turn off Timer/Counter 1.

5	TF0	Timer 0 Overflow Flag. Set by hardware on a Timer/Counter 0 overflow. Cleared by hardware when the PC vectors to the interrupt service routine.
4	TR0	Timer 0 Run Control Bit. Set by user to turn on Timer/Counter 0. Cleared by user to turn off Timer/Counter 0.
3	IE1*	Set by hardware by a falling edge or zero level being applied to external interrupt Pin INT1, depending on bit IT1 state. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
2	IT1*	External Interrupt 1 (IE1) Trigger Type. Set by software to specify edge-sensitive detection (i.e., 1-to-0 transition). Cleared by software to specify level-sensitive detection (i.e., zero level).
1	IE0*	External Interrupt 0 (INT0) Flag. Set by hardware by a falling edge or zero level being applied to external interrupt Pin INT0, depending on bit IT0 state. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
0	IT0*	External Interrupt 0 (IE0) Trigger Type. Set by software to specify edge-sensitive detection (i.e., 1-to-0 transition). Cleared by software to specify level-sensitive detection (i.e., zero level).

*These bits are not used in the control of timer/counter 0 and 1, but are used instead in the control and monitoring of the external INT0 and INT1 interrupt pins.

Timer/Counter 0 and 1 Data Registers
Each timer consists of two 8-bit registers. These can be used as
independent registers or combined to be a single 16-bit register,
depending on the timer mode configuration.

TH0 and TL0
Timer 0 high byte and low byte.
SFR Address = 8CH, 8AH respectively.

TH1 and TL1
Timer 1 high byte and low byte.
SFR Address = 8DH, 8BH respectively.

TIMER/COUNTER 0 AND 1 OPERATING MODES
The following paragraphs describe the operating modes for Timer/Counters 0 and 1. Unless otherwise noted, it should be assumed that these modes of operation are the same for Timer 0 as for Timer 1.

Mode 0 (13-Bit Timer/Counter)
Mode 0 configures an 8-bit Timer/Counter with a divide-by-32 prescaler. Figure 45 shows mode 0 operation.

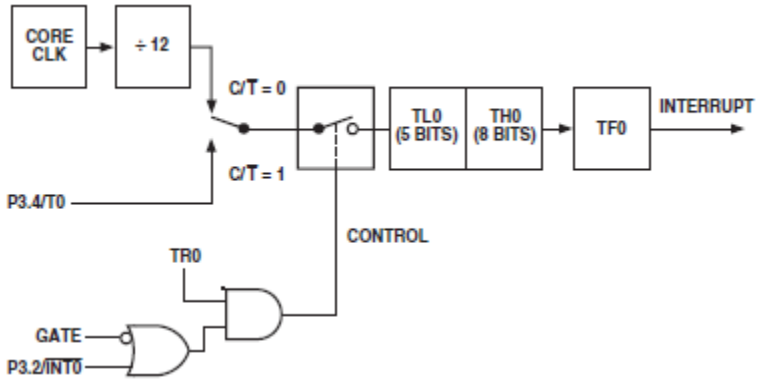


Fig. 45 Timer/Counter 0, Mode 0

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the timer overflow flag TF0. The overflow flag, TF0, can then be used to request an interrupt. The counted input is enabled to the timer when TR0 = 1 and either Gate = 0 or INT0 = 1. Setting Gate = 1 allows the timer to be controlled by external input INT0, to facilitate pulsewidth measurements. TR0 is a control bit in the special function register TCON; Gate is in TMOD. The 13-bit register consists of all eight bits of TH0 and the lower five bits of TL0. The upper three bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers.

Mode 1 (16-Bit Timer/Counter)

Mode 1 is the same as Mode 0, except that the timer register is running with all 16 bits. Mode 1 is shown in Figure 46.

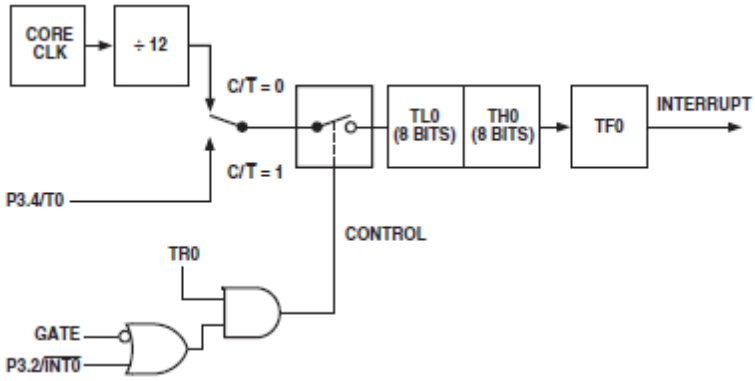


Fig. 46 Timer/Counter 0, Mode 1

Mode 2 (8-Bit Timer/Counter with Autoreload)

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in Figure 47. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is preset by software. The reload leaves TH0 unchanged.

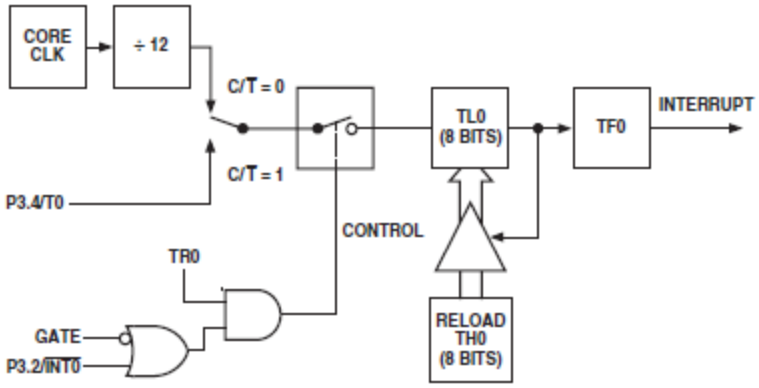


Fig. 47 Timer/Counter 0, Mode 2

Mode 3 (Two 8-Bit Timer/Counters)

Mode 3 has different effects on Timer 0 and Timer 1. Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. This configuration is shown in Figure 50. TL0 uses the Timer 0 control bits: C/T, Gate, TR0, INT0, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt. Mode 3 is provided for applications requiring an extra 8-bit timer or counter.

When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of, and into, its own Mode 3, or can still be used by the serial interface as a baud rate generator. In fact, it can be used in any application not requiring an interrupt from Timer 1 itself.

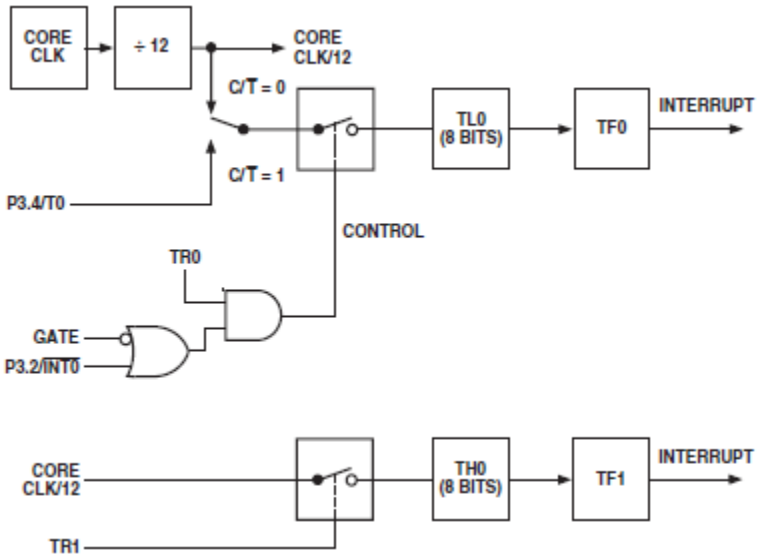


Fig. 48 Timer/Counter 0, Mode 3

d) PROGRAMOWALNE UKŁADY LICZNIKOWE

Mikrokontrolery z rodziny 8051 wyposażone są w programowalne układy licznikowe. Mikrokontroler ADuC831 posiada dwa podstawowe układu o nazwie T0 oraz T1. Oba układy licznikowe są swobodnie konfigurowalne i działają niezależnie od siebie. Dostępne w ADuC831 programowalne układy licznikowe zliczają impulsy w górę. Wyróżnia się dwie podstawowe konfiguracje układów licznikowych: zliczanie impulsów zewnętrznych - licznik (ang. counter), zliczanie impulsów z generatora wzorcowego (sygnał zegarowy) - czasomierz (ang. timer). W drugim przypadku źródłem impulsów jest kwarc zewnętrzny lub wbudowany układ zegara procesora (ang. core clock). Sygnał licznika może być podzielony przez określoną

wartość dzielnika (ang. prescaler). Przykładowo stosując dzielnik równy pięć wartość licznika zostanie zwiększona o jeden po odliczeniu pięciu impulsów. W przypadku ADuC831 sygnał zegarowy jest podzielony przez stałą wartość dzielnika równą 12. Znając wartość dzielnika *p* oraz częstotliwość sygnału generatora f_{osc} można wyznaczyć okres zliczanych impulsów czasomierza τ_c (1).

$$\tau_c = \frac{1}{\frac{f_{osc}}{p}} = \frac{p}{f_{osc}} \tag{1}$$

Zestaw edukacyjny z mikrokontrolerem ADuC831, dostępny podczas zajęć, posiada zewnętrzny rezonator kwarcowy o częstotliwości równej 11,058 MHz.

Liczniki zliczają impulsy z linii P3.4 dla układu T0 oraz P3.5 dla układu T1. Oba liczniki zbudowane są z dwóch 8 bitowych rejestrów THx (bardziej znacząca część) i TLx (mniej znacząca część), gdzie $x = \{0,1\}$. Stan licznika jest określany na podstawie zawartości rejestrów THx oraz TLx. Przepełnienie licznika następuje po przekroczeniu maksymalnej wartości w danym trybie. Przykładowo w trybie 16 bitowym zliczenie kolejnego impulsu do wartości 0xFFFF spowoduje przepełnienie licznika. Przepełnienie licznika powoduje wyzerowanie jego wartości oraz ustawienie bitu TFx informującego o tym zdarzeniu. Jeżeli zostały ustawione zgody na przerwania to program przejdzie do funkcji obsługującej przerwanie.

Zliczenie *N* impulsów z generatora wzorcowego o okresie τ_c odpowiada odmierzeniu czasu τ (2). Odmierzenie określonego czasu τ_{zad} sprowadza się do wyznaczenia liczby zliczanych impulsów *N* przy określonym okresie τ_c generatora wzorcowego (3). Wynik należy zaokrąglić. Pamiętając, że dostępne liczniki zliczają w górę należy określić wartość początkową licznika *T*. Od wartości maksymalnej możliwej do zliczenia w danym trybie należy odjąć liczbę impulsów *N* (4). Licznik może maksymalnie zliczyć 2^n impulsów, gdzie *n* oznacza łączną liczbę bitów przeznaczonych do przechowania wartości zliczania. Obliczoną wartość *T* należy wpisać do odpowiednich rejestrów THx oraz TLx.

$$\tau = \tau_c \cdot N \tag{2}$$

$$N = \frac{\tau_{zad}}{\tau_c},\ gdzie\ N \in \mathbb{N} \tag{3}$$

$$T = 2^n - N,\ gdzie\ T \in \mathbb{N} \tag{4}$$

Przykład 1.

Czasomierz ma odliczyć czas równy $\tau_{zad} = 1\,ms$. Częstotliwość sygnału generatora wynosi $f_{osc} = 12\ \mathrm{MHz}$, natomiast dzielnik jest równy $p = 12.$ Znając wartość dzielnika *p* oraz częstotliwość sygnału generatora f_{osc} wyznaczono okres zliczanych impulsów czasomierza $\tau_c = \frac{12}{12 \cdot 10^6} s = 1 \cdot 10^{-6} s = 1\mu s$. Odmierzenie określonego czasu τ_{zad} sprowadza się do wyznaczenia liczby zliczanych impulsów $N = \frac{1\,ms}{1\,\mu s} = 1000$. W omawianym przykładzie liczba $N \in \mathbb{N}$. Początkowo założono *n=8* bitów przeznaczonych do przechowania wartości zliczania uzyskując $T = 2^8 - 1000 = -744$. Ujemna wartość *T* oznacza przyjęcie za małego rozmiaru rejestru przeznaczonego do zliczania. Problem można rozwiązać na dwa sposoby: zmieniając konfigurację długości rejestru na *n=16* lub dodając programowe zliczanie krótszych okresów czasu np. czterokrotne odmierzenie czasu $250\,\mu s$. Przyjmując długości rejestru równą *n=16* uzyskano

$T = 2^{16} - 1000 = 65536 - 1000 = 64536 = 0\text{xFC18}$. Obliczoną wartość $T = 0\text{xFC18}$ należy wpisać do odpowiednich rejestrów THx=0xFC oraz TLx=0x18, gdzie $x = \{0,1\}$.

LITERATURA

1. ADUC831 datasheet and product info | Precision Analog Microcontroller: 1.3MIPS 8052 MCU + 62kB Flash + 8-Ch 12-Bit ADC + Dual 12-Bit DAC | Analog Microcontrollers | Analog Devices [online]. [udostępniono 19.11.2014]. Pobrano: http://www.analog.com/en/processors-dsp/analog-microcontrollers/aduc831/products/product.html.
2. *ADuC831 MicroConverter, 12-Bit ADCs and DACs with Embedded 62 kBytes Flash MCU Data Sheet, (Rev. 0) - ADUC831.pdf* [online]. s.l.: s.n. [udostępniono 19.11.2014]. Pobrano: http://www.analog.com/static/imported-files/data_sheets/ADUC831.pdf.

IV. SCENARIUSZ DO ZAJĘĆ

a) ŚRODKI DYDAKTYCZNE

Sprzętowe:
- komputer,
- zestaw edukacyjny z mikrokontrolerem ADuC831.

Programowe:
- zintegrowane środowisko programistyczne języka C dla mikrokontrolerów (Keil uVision).

b) PRZEBIEG ZAJĘĆ

1. Ustaw układ licznikowy T0 w trybie licznika 16 bitowego.
   a. Przygotuj makroinstrukcje do konfiguracji układu licznikowego T0.
   b. Napisz fragment kodu w pętli głównej sprawdzający stan flagi.
      i. Skonfiguruj układ licznikowy oraz wartość początkową licznika.
      ii. Co 5 impulsów zmień wartość zmiennej globalnej na przeciwną.
      iii. Pamiętaj o ponownym ustawieniu wartości początkowej licznika oraz kasowaniu flagi.
      iv. Sprawdź działanie programu w symulatorze w trybie krokowym.
   c. Napisz funkcję obsługującą przerwanie od przepełnienia układu licznikowego T0.
      i. Ustaw odpowiednie zgody na przerwania.
      ii. Co 5 impulsów zmień wartość zmiennej globalnej na przeciwną.
      iii. Pamiętaj o ponownym ustawieniu wartości początkowej licznika. Czy flagę należy kasować?
      iv. Sprawdź działanie programu w symulatorze w trybie krokowym.
2. Ustaw układ licznikowy T1 w trybie czasomierza 16 bitowego.
   a. Przygotuj makroinstrukcje do konfiguracji układu licznikowego T1.
   b. Przygotuj definicje oraz makroinstrukcje do wyznaczenia wartości początkowej licznika *T*, na podstawie żądanego czasu do odmierzenia τ_{zad}.
   c. Napisz fragment kodu w pętli głównej sprawdzający stan flagi.
      i. Skonfiguruj układ licznikowy oraz wartość początkową czasomierza.
      ii. Co 20 impulsów generatora wzorcowego zmień wartość zmiennej globalnej na przeciwną.

iii. Pamiętaj o ponownym ustawieniu wartości początkowej czasomierza oraz kasowaniu flagi.
iv. Sprawdź działanie programu w symulatorze w trybie krokowym.
v. Sprawdź działanie programu w symulatorze w trybie ciągłym. Wykorzystaj dostępny analizator sygnałów do rysowania zmiany wartości zmiennej globalnej w trakcie trwania symulacji.

d. Napisz funkcję obsługującą przerwanie od przepełnienia układu licznikowego T1.
i. Ustaw odpowiednie zgody na przerwania.
ii. Co 20 impulsów generatora wzorcowego zmień wartość zmiennej globalnej na przeciwną.
iii. Pamiętaj o ponownym ustawieniu wartości początkowej licznika. Czy flagę należy kasować?
iv. Sprawdź działanie programu w symulatorze w trybie krokowym.

3. Proste wykorzystanie czasomierza.
a. Cyklicznie zmieniaj stan diody na przeciwny w funkcji obsługującej przerwanie.
b. Okres zmian ustala prowadzący (np. 1 s).
c. Przetestuj działanie programu na zestawie edukacyjnym.
4. Proste wykorzystanie czasomierza.
a. Cyklicznie zmieniaj stan diody na przeciwny w funkcji obsługującej przerwanie.
b. Cyklicznie zmieniaj stan drugiej diody na przeciwny w pętli głównej programu.
c. Cyklicznie co zadany czas zwiększ *k* o 0,1 oraz oblicz $k \cdot \sqrt{k}$, gdzie *k* zmienia się z zakresie od 1 do 10.
d. Okres obliczeń ustala prowadzący (np. 1 ms).
e. Sprawdź działanie programu. Jaki wpływ na działanie programu ma umieszczenie obliczeń w pętli głównej a jaki w funkcji obsługującej przerwanie?
5. Złożone programy wykorzystujące oba układy licznikowe.
a. Wyznaczyć liczbę impulsów zewnętrznych w zadanym okresie czasu.
b. Wyznaczyć czas między kolejnymi impulsami zewnętrznymi.
c. Sprawdź działanie obu programów w symulatorze.