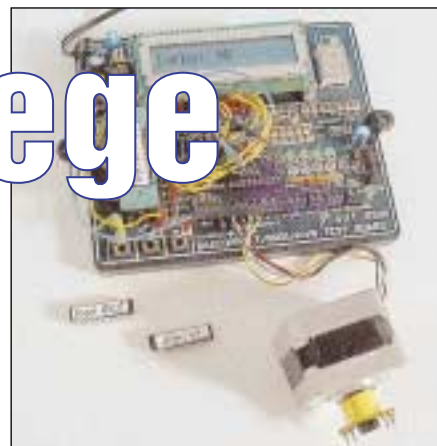


# BASCOM College



## Ćwiczenie 5



## Pomiary częstotliwości i czasu

Dzisiejszy wykład miał wyjątkowo wakacyjny i wypoczynkowy charakter, a zatem i ćwiczenie nie powinno okazać się zbyt trudne i wyczerpujące. Chciałbym nawiązać w nim do koncepcji lansowanej przeze mnie w poprzednim ćwiczeniu: idei stosowania naszej płytki testowej i odpowiednio zaprogramowanego procesora jako **UPPW**, czyli Uniwersalnego Programowalnego Przyrządu Warsztatowego. Dzisiaj pokażę Wam, jak w ciągu kilku - kilkunastu minut możemy zbudować, a właściwie zaprogramować prosty miernik częstotliwości, o parametrach zupełnie zaspokajających potrzeby przeciętnego hobbyisty.

Chciałbym jeszcze zwrócić Waszą uwagę na fakt, że nasza "baza sprzętowa" uległa ostatnio rozszerzeniu. Opisana w poprzednim numerze EdW uniwersalka do procesorów 'X051 może także znaleźć zastosowanie do budowy przyrządów pomiarowych. Opisany w dzisiejszym ćwiczeniu miernik częstotliwości możemy zatem umieścić na płytce testowej i po przetestowaniu przenieść na płytkę uniwersalną. Uzyskamy w ten sposób prosty, przenośny przyrząd pomiarowy, do którego w miarę potrzeby będziemy mogli "doprogramowywać" kolejne dodatkowe funkcje.

### Miernik częstotliwości w 10 minut

Wiemy już na tyle dużo o timerach i przerwaniach, aby pokusić się o skonstruowanie, a właściwie zaprogramowanie prostego miernika częstotliwości, którego parametry w głównej mierze zależą będą od stopnia komplikacji programu. Nie obiecujemy jednak sobie zbyt wiele: na naszej płytce testowej będziemy mogli zbudować miernik o zakresie pomiarowym nie przekraczającym 500kHz (z typowym kwarcem 11MHz), a w najlepszym wypadku, po wymianie kwarcu, do 1MHz. Jednak już ta pierwsza wartość może okazać się w praktyce hobbyistycznej zupełnie wystarczająca.

Za chwilę zaprogramujemy sobie aż trzy mierniki częstotliwości: najprostszy, o nie najlepszych parametrach, ale za to do wykonania dosłownie w minutę, bardziej rozbudowany, o zwiększonej dokładności i "profesjonalny", maksymalnie wykorzystujący możliwości "gołego" procesora 89C2051.

Napiszmy zatem program pierwszego, najprostszego miernika częstotliwości:

Działanie tak napisanego programu nie wymaga chyba komentarza. Po skompilowaniu i zapisaniu programu w pamięci procesora montujemy na naszej płytce testowej układ pokazany na **rysunku 1**. Poza procesorem i wyświetlaczem wykorzystujemy tu tylko

```

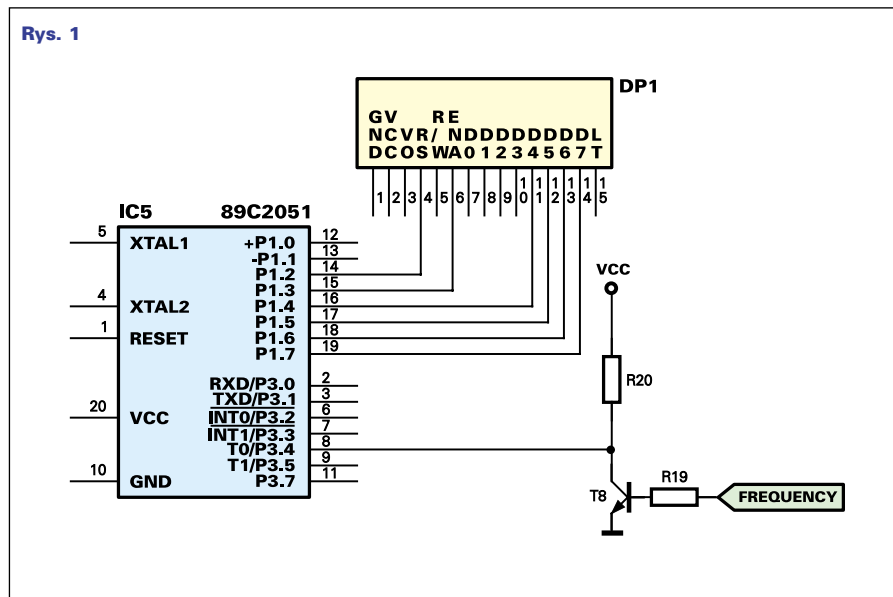
Config Timer0 = Counter , Gate = External , Mode = 1
'timer0 został skonfigurowany jako licznik 16-bitowy, sterowany impulsami zewnętrznymi 'podawanymi
                                                                    na wejście T0

Config Lcd = 16 * 1a                               'określenie typu wyświetlacza LCD
Dim Frequency As Word                             'deklaracja zmiennej FREQUENCY jako liczby 16-bitowej
Start Counter0                                    'uruchomienie licznika

Do                                                  'początek pętli programowej
Cls                                                'czyszczenie ekranu wyświetlacza
Lcd "FRQ=: " ; Frequency ; " Hz"                 'wyświetl liczbę zliczonych w ciągu 1 sekundy impulsów 'wejściowych
Frequency = 0
Counter0 = 0                                       'zawartość licznika 0 wynosi teraz 0
Start Counter0                                    'początek zliczania impulsów wejściowych
Wait 1                                             'zaczekaj 1 sekundę, czyli odmierz czas bramkowanie miernika
Stop Counter0                                     'zatrzymaj licznik 0
Frequency = Counter0                               'zmienna FREQUENCY przyjmuje wartość licznika 0, czyli określa liczbę 'zliczonych
                                                                    impulsów wejściowych

Loop                                                'zamknięcie pętli programowej
    
```

Rys. 1



jeden element z zasobów naszej płytki: tranzystor NPN, który będzie pełnił rolę bufora oddzielającego wejście procesora od badanego układu. Ponadto, zastosowanie tranzystora umożliwi nam pomiar przebiegów o poziomie innym niż poziomy TTL. Następnie do wejścia układu dołączmy źródło impulsów prostokątnych, najlepiej o regulowanej częstotliwości, i obserwujemy wyświetlacz LCD. Co sekundę pojawiać się będzie na nim zaktualizowany odczyt zmierzonej wartości.

Tak zaprogramowany miernik ma dwie wady: małą dokładność wynikającą z zastosowania do odmierzenia czasu bramkowania polecenia WAIT, w założeniu przeznaczonych do mniej precyzyjnych zastosowań, i ograniczenie maksymalnej mierzonej częstotliwości do 65535 Hz. To drugie ograniczenie wynika z maksymalnej pojemności 16-bitowego timera i jest szczególnie dokuczliwe, ponieważ nie sygnalizuje nam, że licznik został przepełniony i rozpoczął zliczanie impulsów od początku. Wynik np. "12000Hz" może zatem oznaczać równie dobrze 12000Hz, jak i  $12000 + 65536$ Hz, jak również  $12000 + 65536 \times [\text{liczba przepełnień licznika}]$  Hz. Mamy dwie możliwości usunięcia tej wady: albo dopisać do programu następującą kilka linijek:

```
On Timer0 Overflow 'w przypadku przepełnienia
licznika skocz do podprogramu OVERFLOW
```

oraz podprogramu:

```
Overflow:
Cls 'wyczyść ekran LCD
Lcd "Overflow!" 'komunikat o przekroczeniu
zakresu
Wait 1 'zaczekaj chwilę przeznaczoną
na odczytanie komunikatu
Return 'powrót do programu głównego
```

albo też rozszerzyć zakres pomiarowy naszego przyrządu.

Z rozszerzeniem zakresu nie będziemy mieli najmniejszego problemu, oczywiście w granicach możliwości stosowanego procesora i kwarcu. Maksymalna częstotliwość, jaką możemy podać na wejście timera procesora '51 wynosi bowiem:  $F_{osc}/24$ , czyli w przypadku typowego kwarcu 11,059MHz nieco ponad 460KHz (i to przy założeniu, że przy tej maksymalnej częstotliwości badany przebieg ma wypełnienie 1/2). Niewiele, ale nawet po wymianie kwarcu nie osiągniemy nigdy wyniku lepszego od 1,5MHz. Wracajmy jednak do programowego rozszerzenia zakresu pomiarowego naszego miernika, a o możliwościach sprzętowych wspomnimy jeszcze za chwilę.

Wiemy już, że ograniczenie zakresu pomiarowego naszego miernika wynika z pojemności licznika TIMER, wbudowanego w strukturę procesora. Wiemy także, że w momencie przepełnienia licznik ten generuje sygnał zgłoszenia przerwania, który już

wykorzystaliśmy do sygnalizowania przekroczenia zakresu. Zmierimy zatem podprogram obsługi tego przerwania, dodajmy do progra-

```
Dim Counts As Byte
```

mu jeszcze jedną zmienną i zobaczymy, co z tego wyniknie.

Deklarujemy na początku programu zmienną pomocniczą, którą możemy nazwać TEMP

Wiemy, że mierzona wartość będzie teraz mogła być większa niż 65535, co powoduje konieczność zwiększenia maksymalnej wartości zmiennej FREQUENCY. Deklarujemy ją zatem jako LONG, czyli liczbę 32-bitową, o maksymalnej wartości

```
Dim Frequency As Long
```

2147483647, czyli nawet znacznie większą, niż potrzebujemy.

```
Overflow:
Incr Temp 'wystąpienie przepełnienia
licznika powoduje zwiększenie wartości TEMP o 1
Return
```

Następnym krokiem będzie przeróbka podprogramu obsługi przerwania TIMER0:

```
Frequency = 65536 * Temp 'częstotliwość równa jest pojemności licznika * liczba przepełnień
Temp = Counter0 'wykorzystujemy ponownie zmienną TEMP, nadając jej wartość 'ostatniego wyniku zliczania
licznika TIMER0
Frequency = Frequency + Temp 'ostateczne obliczenie wartości FREQUENCY
```

oraz dodanie kilku linijek, w których obliczana będzie wartość zmierzonej częstotliwości:

A zatem druga wersja naszego programu miernika częstotliwości będzie miała postać

```
Config Timer0 = Counter , Gate = External , Mode = 1
Config Lcd = 16 * 1
Dim Frequency As Long
Dim Temp As Byte
Enable Interrupts
Enable Timer0
On Timer0 Overflow
Cls

Do
Cls
Lcd "FRQ=" ; Frequency ; " Hz"
Frequency = 0
Counter0 = 0
Temp = 0
Start Counter0
Wait 1
Stop Counter0
Frequency = 65536 * Temp
Temp = Counter0
Frequency = Frequency + Temp
Loop

Overflow:
Incr Temp
Return
```

przedstawioną na poniższym listingu (pozostawimy tym razem program bez komentarzy):

Pozostała nam zatem tylko jedna, ale dość istotna wada: nieszczęsne WAIT 1. Polecenie to nie nadaje się do dokonywania precyzyjnych pomiarów, o czym zresztą zostaliśmy lojalnie uprzedzeni w instrukcji - helpie. A zatem, jeżeli chcemy zbudować miernik o przyzwoitych parametrach, musimy dokonać kolejnej rozbudowy programu.

Z pewnością przypominać sobie, jak na poprzedniej lekcji bawiliśmy się timerami i zbudowaliśmy sobie eksperymentalny model sekundnika do zegarów mikroprocesorowych. Najwyższy czas, aby wykorzystać tamte doświadczenia i zastosować zdobytą wiedzę w praktyce. Sekundnik będzie jednak nam potrzebny nie do zegara, ale do bramkowania naszego miernika częstotliwości.

Jako "silnik" napędzający nasz sekundnik wykorzystamy drugi, wolny do tej pory Ti-

```
Config Timer1 = Timer , Gate = Internal , Mode = 1
```

mer - TIMER1, pracujący w trybie 1. A więc piszemy na początku programu:

Tak skonfigurowany timer będzie pracował jako licznik 16-bitowy, niestety bez możliwości automatycznego ładowania. A teraz musimy trochę policzyć, kalkulatory w dłoń!

Naszym zadaniem jest dokładne odmierzenie odcinka czasu, równego jednej sekundzie. Jak punkt wyjścia przyjmiemy częstotliwość podstawową pracującego w systemie rezonatora kwarcowego: 11 059 200Hz. Wiemy, że częstotliwość generowana przez wbudowany w procesor 89C2051 generator jest wstępnie dzielona przez 12. A zatem częstotliwość taktująca naszego procesora wyniesie  $11059200/12 = 921600$ Hz. Liczba ta znacznie przekracza pojemność zastosowanego timera, który wobec tego zostanie zmuszony do kilkukrotnego zliczania w ciągu sekundy. Musimy teraz wybrać możliwie największą liczbę, przez którą musimy podzielić tę częstotliwość, aby rezultat działania był liczbą całkowitą. Po długotrwałych próbach i rozgrzaniu kalkulatora do czerwoności, doszedłem do wniosku, że liczbą tą jest 15, to znaczy  $921600/15 = 61440$ . Bardzo dobry wynik, wykorzystywać będziemy prawie całą pojemność timera!

**Reasumując: aby odmierzyć 1 sekundę, timer musi 15 razy odliczyć 61440 impulsów.**

Wynika z tego, że aby osiągnąć przepełnienie timera i wygenerowanie przerwania we właściwym czasie, timer musi za każdym razem rozpocząć zliczanie od wartości 4096 ( $65536 - 61440 = 4096$ ). Teoretycznie!

Nie zapominajmy bowiem, że na zatrzymanie, przeładowanie i ponowne uruchomienie timera procesor potrzebuje także trochę czasu. Z obliczeń i z doświadczeń przeprowadzonych w symulacji programowej wynika, że na te operacje procesor "zużyje" aż 54 takty zegarowe. A zatem musimy dać timerowi "fory":  $4096 + 54 = 4150$ .

Ostateczna postać trzeciej, finalnej wersji naszego programu została pokazana na poniższym listingu.

```

Config Timer0 = Counter , Gate = External , Mode = 1
Config Timer1 = Timer , Gate = Internal , Mode = 1
Config Lcd = 16 * 1
Dim Frequency As Long
Dim Temp As Byte
Dim Counts As Byte
Declare Sub 1sekunda
On Timer1 Timer1_int
Enable Interrupts
Enable Timer0
Enable Timer1
Cls

Do
Cls
Lcd "FRQ=" ; Frequency ; " Hz"
Frequency = 0
Counter0 = 0
Temp = 0
Call 1sekunda
Stop Counter0
Frequency = 65536 * Temp
Temp = Counter0
Frequency = Frequency + Temp
Loop

Sub 1sekunda
Counts = 0
Counter1 = 4150
Start Timer1
Start Timer0

Do
If Counts = 15 Then
Stop Timer1
Stop Timer0
Exit Do
End If
Counts = Counts + 1
Loop

End Sub

Timer1_int:
Counter1 = 4150
Start Timer1
Incr Counts
Return
    
```

```

Sub 1sekunda
Counts = 0
Counter1 = 4150
Start Timer1
Start Timer0

Do
If Counts = 15 Then
Stop Timer1
Stop Timer0
Exit Do
End If
Counts = Counts + 1
Loop

End Sub

Timer1_int:
Counter1 = 4150
Start Timer1
Incr Counts
Return
    
```

Listing pełnej wersji programu nie został opatrzony komentarzami. Sądzę bowiem, że nauczyliście się wystarczająco dużo, aby bez mojej pomocy zrozumieć treść tego prostego programiku, zajmującego w pamięci procesora tylko nieco więcej niż 850 bajtów.

Podsumujmy zatem dzisiejsze osiągnięcia: zbudowaliśmy, a właściwie zaprogramowaliśmy, prosty miernik częstotliwości o przyzwoitych parametrach. Nie kosztowało to nas złamanego grosza, a nasz miernik możemy przechowywać na dysku komputera i w razie potrzeby programować procesor i łączyć kilka kabelek na płytce testowej. Czyba było warto? Pomyślcie tylko, ile pracy i pieniędzy kosztowałby taki

przyrząd realizowany metodami z początku lat 80., jakie stosowaliście do tej pory! A przecież to dopiero początki, dopiero nabieramy rozpędu, który będzie rosł w postępie geometrycznym!

Skończmy jednak z tym samouwielbieniem i popatrzmy jeszcze na nasz miernik krytycznym okiem. Jest dość dokładny, ale zakres pomiarowy mizerny. Nawet jeżeli wymienimy kwarc (i dokonamy stosownych przeliczeń w programie) na 24MHz, to i tak wiele nie uzyskamy. Można wprowadzić zastosować preskaler, ale naszym założeniem była budowa miernika częstotliwości z wykorzystaniem wyłącznie elementów dostępnych na płytce testowej. Co za tym można zrobić? Można, ale powiem Wam to w tajemnicy, bo inaczej znowu oberwę za wprowadzanie bałaganu do BASCOM College. Można wymienić procesor na ... np. AT90S2313, nowoczesny chip zrealizowany w technologii RISC, który bez problemu "połknie" częstotliwość do 5MHz. BASCOM AVR Demo dostępny jest w Internecie pod adresem [www.ep.com.pl](http://www.ep.com.pl), a nasz program wymagałby tylko minimalnych przeróbek (to właśnie jedna z zalet języków wysokiego poziomu). Ale pamiętajcie, że powiedziałem Wam o tym w sekrecie, bo procesorami AVR zajmiemy się później.

**Zbigniew Raabe**

e-mail: [zbigniew.raabe@edw.com.pl](mailto:zbigniew.raabe@edw.com.pl)

Konsultacje: **Sławomir Surowiński**

e-mail: [slawomir.surowinski@ep.com.pl](mailto:slawomir.surowinski@ep.com.pl)

## Uwaga! Nadal można zgłosić się do BASCOM College!

### Zgłoszenie uczestnictwa w BASCOM College

Tak, zgłaszam chęć uczestnictwa w Szkole Programowania Mikroprocesorów BASCOM College

nazwisko .....

imię ....., wiek .....

ulica .....

kod ....., poczta .....

Przyslijcie mi zaległe artykuły z:

EdW 2/2000,  EdW 3/2000,  EdW 4/2000,

EdW 5/2000,  EdW 6/2000,  EdW 7/2000,

Jestem prenumeratorem EdW, mój numer prenumeratora:

.....

Zamierzam wkrótce zostać prenumeratorem EdW,

Zamawiam płytki emulatora (AVT-2501),

programator (AVT-2502),

płytkę testową (AVT-2500),

oraz płytę CD-ROM z programem

ww. płytki zamawiam:

za darmo, gdyż jestem (wkrótce będę) prenumeratorem EdW,

za pobraniem pocztowym (39zł 60gr + koszt wysyłki 12,5zł)

Podpis ....., Data .....

Niniejsze „zgłoszenie „, można przesłać:

- pocztą na adres: AVT, ul. Burleska 9, 01-939 Warszawa

- faksem na numer: (0-22) 835-67-67 lub (0-22) 676-89-86

- e-mailem na adres: [prenumerata@avt.com.pl](mailto:prenumerata@avt.com.pl)

Uwaga, wszyscy studenci BASCOM College będący prenumeratorem EdW otrzymają 20% rabatu przy zakupie kitów AVT2500, AVT2501, AVT2502 w wersji B.