

| Systemy wbudowane Laboratorium |  |                             |  |
|--------------------------------|--|-----------------------------|--|
| Grupa:<br><b>C</b>             | Temat:<br><b>Zastosowanie licznika-czasomierza Timer0 do generowania stałych odcinków czasu.</b> |                             |  |
| Data:<br><b>22.04.2021</b>     | Wykonał:<br><b>Maciej Bujalski</b>   |                             |  |
| Godzina:<br><b>12.40</b>       | <b>II rok Informatyka Stosowana</b>  | Ocena i uwagi prowadzącego: | Prowadzący:<br><b>Witold Kozłowski</b> |

## 1. Cele

- i. Generowanie przerwań przy przepełnieniu Timer0
- ii. Przykład generowania sygnału PMW
- iii. Regulacja jasności świecenia diody LED za pomocą sygnału PMW

## 2. Co robiliśmy / Opis:

Licznik Timer0 jest 8 bitowy co oznacza, że może zliczyć tylko 256 impulsów. Może on zliczać impulsy zegara taktującego procesor bezpośrednio lub przez prescaler. Zliczanie można dowolnie zatrzymywać i wznowiać oraz wpisywać wartości do licznika. Prescaler może mieć ustawione wartości 1,8,64,256,1024 co rzutuje na pomiar (poprzez częstotliwość oscylatora w tym wypadku kwarcowego oraz RC).

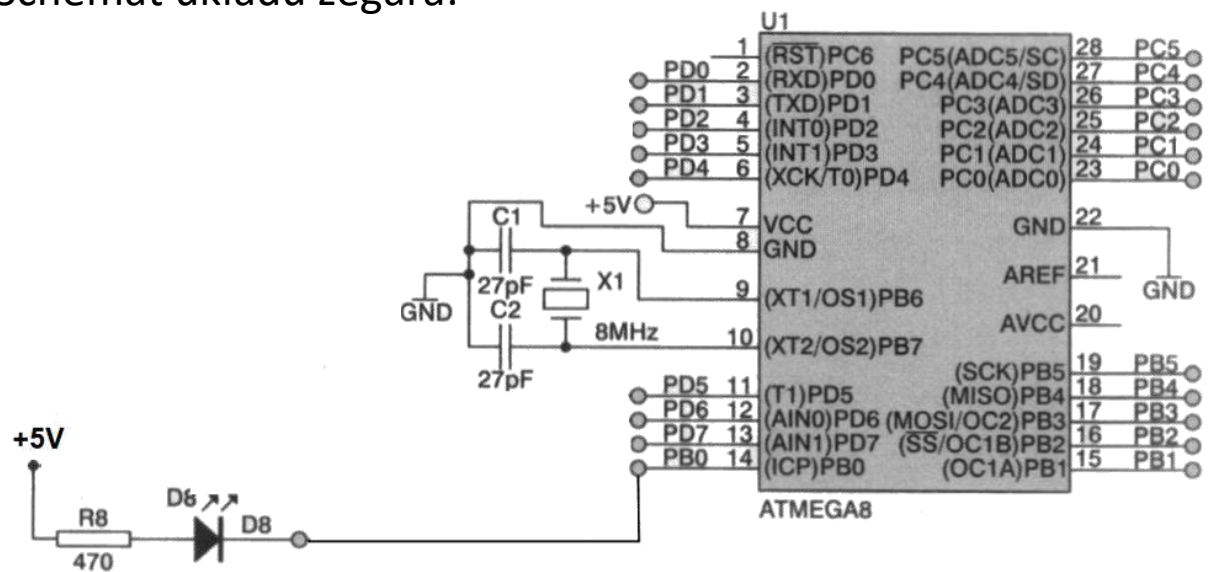
Wygenerowaliśmy w ten sposób odcinek (prawie) 1s. Niestety pomiary z użyciem rezonatora RC nie były na tyle dokładne aby mógł być sensownie używany, ponieważ różnica czasu wynosiłaby mniej więcej 3 dni w ciągu roku. W tym wypadku okres całego układu wynosił ~1,98 sekundy.

Następnie wykorzystaliśmy rezonator kwarcowy, gdzie 1s to faktycznie 1s, a okres całego układu wynosił 1.00 s co daje częstotliwość 0,5 Hz.

Następnie wzięliśmy pod lupę proces generowania wypełnienia sygnału PWM. Mogliśmy za jego pomocą kontrolować przepływ prądu między stanami UP i DOWN odpowiadającymi kolejno 1 i 0 logicznemu.

Naszym zadaniem było ustawienie takich wartości prescalera oraz takie ustawienie TIMERA aby uzyskać okres 512us czyli  $f$  bliskiego 2 kHz. Jako następne zadanie mieliśmy ustawienie diody w taki sposób, aby emisja światła była rosnąca, a następnie malejąca.

### 3. Schemat układu zegara:



### 4. Kod programu:

```
$regfile = "m8def.dat"
$crystal = 8000000

Config PINB.0 = Output
Config TIMER0 = Timer, Prescale = 64

On TIMER0 Odmierz_1s
Dim Licz_8ms As Byte

Enable Interrupts
Enable TIMER0
Load TIMER0 = 150

Do
Loop

End

Odmierz_1s:

Load TIMER0 , 150
Incr Licz_8ms

If Licz_8ms = 125 Then

Licz_8ms = 0
Toggle PORTB.0

End If
Return
```

W pierwszych dwóch liniach są dyrektywy dla procesora oraz częstotliwość oscylatora.  
W trzeciej linii konfigurujemy port B jako wyjście  
W czwartej linii konfigurujemy Timer0.

Następnie mamy nieskończoną pętlę, w której ciele znajduje się funkcja Set, która włącza portB oraz Funkcja Reset która wyłącza portB .

#### Niepełna jedna sekunda:

CurA= -512.0ms  
CurB=480ms  
 $\Delta X=992.0\text{ms}$   
 $1/\Delta X=1.008\text{Hz}$

#### Pełny okres 1s:

CurA= -688.0ms  
CurB=480ms  
 $\Delta X=992.0\text{ms}$   
 $1/\Delta X=1.008\text{Hz}$

#### Zadanie z PWM:

CurA= -512.0ms  
CurB=4.000ms  
 $\Delta X=512.0\text{ms}$   
 $1/\Delta X=1.953\text{Hz}$

Pwm1a odpowiada za % wypełnienia czyli przy 8 bitach PWM = 127 to 50% wypełnienia

Aby osiągnąć zadowalający nas wynik zapisaliśmy poniższe ustawienia:

PWM=8

```
$regfile = "m8def.dat"  
$crystal = 8000000
```

PRESCALE = 8

```
Config Pinb.1 = Output  
Config PINB.2 = Output
```

Pwm1a = 178

```
Config Timer1 = Pwm , Pwm = 8, Compare A Pwm = Clear UP, Compare B Pwm = Clear DOWN, Prescale = 8
```

Sterowanie diodą:

```
dim A as word  
DO  
for A=1 to 255  
Pwm1a = A  
Pwm1b = A  
WAITMS 4  
next A  
for A=255 to 1 STEP -1  
Pwm1a = A  
Pwm1b = A  
WAITMS 4  
next A  
Loop  
End
```

## 5. Wniosek:

Aby czas odmierzony został poprawnie należało zmienić linię Timer0 , 250 na Timer0 = 250, gdyż inaczej program powodował opóźnienia dodatkowo dowiedzieliśmy się, że rezonator kwarcowy daje dużo dokładniejsze rezultaty. W następnym programie użyliśmy Timera jako wypełnienie sygnału(PWM). Dowiedzieliśmy się do czego można wykorzystać sygnał PWM tj. np. Kontrolowanie podświetlania wyświetlacza LCD.