

Sprawozdanie nr. 2

Mateusz Kojro

1 Opis cwiczenia

Celem tego cwiczenia jest poznanie sposobow uzywania i wykorzystanie licznika jako ukladu mierzacego czasu (Timer0) ale rowniez przyklad generowania sygnalu PWM (Timer1) Timery w mikrokontrolerze dzialaja w zasadzie licznikow. Wykorzystywany przez nas mikrokontroler posiada 2 rozne timery 8 i 16 bitowy sa one wiec w stanie odliczac odpowiednio do 256 i 65 536 w rownych odstepach czasu. Posiadajac taki timer jestesmy w stanie wygenerowac sygnal PWM (Pulse width modulation) ktory dziala na zasadzie generowania przebiegu prostokatnego o zmiennych dlugosciach wysylania 0 i 1 sygnal taki umozliwia np. stopniowe przyciemnianie diody LED

2 Wykorzystane polecenia jezyka BASCOM

- `Config Timer0` - Konfiguracja pracy Timer0
- `Start` | `Stop` - Umozliwia zatrzymanie i wystartowanie licznika
- `Counter0 = value` - Ustawia poczatkowa wartosc licznika
- `Load Timer0 number` - Wartosc po ktore licznik sie przepelni
- `On Interrupt` - Konfiguruje dzialania wykonywane podczas ogloszenia przerwania
- `Enable` | `Disable Interrupt` - Ustawienie zgłaszania przerwan podczas pracy mikrokontrolera

3 Wykorzystywane przyrzady

- Oscyloskop cyfrowy

4 Przeprowadzone doswiadczenia

Gdy dokonalismy pomiaru okresu przerwan wbudowanego timera teoretycznie ustawionego na okers 1 sekundy otrzymalismy okres zmierzony za pomoca oscyloskopu rowny 968ms. Jak latwo obliczyc w skali roku skonstruowany w ten

sposob zegar mylil by sie o ponad 11 dni. Aby otrzymac wyniki blizsze prawdy nalezy wykorzystac zewnetrzny rezonator kwarcowy.

Listing 1: Program odmierzajacy 1s

```

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

Config Pinb.0 = Output
Config Timer0 = TIMER, Prescale = 256
256

On Timer0 Odmierz_1s

Dim Licz_8ms As Byte

Enable Interrupts
Enable Timer0
Load Timer0 = 250

Do
Loop
End

Odmierz_1s:
    Load Timer0 =250
    Incr Licz_8ms
    If Licz_8ms = 125 Then
        Licz_8ms = 0
        Toggle Portb.0
    End If
Return

```

Aby wygenerowac sygnal PWM o czestotliwosci 7.82 kHz i wypelnionego w 60%

Czestotliwosc:

$$F_c = \frac{\frac{8 \cdot 10^6 Hz}{1}}{1022} = 0.00782 * 10^6 Hz = 7.82 KHz \quad (1)$$

Wypelnienie:

$$k_w = 6 * x = 306.6 \quad (2)$$

$$\begin{aligned} 511 &\leftrightarrow 100\% \\ x &\leftrightarrow 10\% \end{aligned} \quad (3)$$

mamy wiec:

$$x = 511 * \frac{10}{100} = 51.1 \quad (4)$$

Listing 2: Kod programu generujacego sygnal PWM

```
$regfile = "m8def.dat"  
$crystal = 8000000  
Config Pinb.1 = output  
Config Timer1 = PWM,  
PWM = 8,  
Compare A PWM = Clear Up,  
Compare B PWM = Disconnect,  
Prescale = 256  
PWM1a = 127.5  
End
```
