

Systemy wbudowane Laboratorium			
Grupa: C	Temat: Wyświetlacze		
Data: 06.05.2021	Wykonał: Maciej Bujalski		
Godzina: 12.40	II rok Informatyka Stosowana	Ocena i uwagi prowadzącego:	Prowadzący: Witold Kozłowski

1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z wyświetlaczami 7-segmentowymi oraz wyświetlaczami LCD. W ćwiczeniu posługiwaliśmy się czterema wyświetlaczami 7-segmentowymi, które przez odpowiednie zaprogramowanie mikrokontrolera wyświetlały odpowiednie liczby lub znaki w danej kolejności. Wyświetlacze LCD zaś wyświetlały pewne łańcuchy znakowe, które mogły się po tym ekranie poruszać.

2. Pomiary

Program 1

Program 1 miał za zadanie wyświetlić na pojedynczym wyświetlaczu liczby od 1 do 9, następnie liczbę 0, następnie napis HELP litera po literze. Program za pomocą instrukcji Do Loop wykonywał się w pętli. Czas przez jaki każdy znak był wyświetlany na wyświetlaczu wynosił 1 sekundę. Kod programu widać poniżej.

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

Config PORTB = Output

Do
PORTB = &B00000110
Vaitas 1000
PORTB = &B01011011
Vaitas 1000
PORTB = &B01001111
Vaitas 1000
PORTB = &B01100110
Vaitas 1000
PORTB = &B01101101
Vaitas 1000
PORTB = &B01111101
Vaitas 1000
PORTB = &B00000111
Vaitas 1000
PORTB = &B01111111
Vaitas 1000
PORTB = &B01101111
Vaitas 1000
PORTB = &B00111111
Vaitas 1000
PORTB = &B01110110
Vaitas 1000
PORTB = &B01111001
Vaitas 1000
PORTB = &B00111000
Vaitas 1000
PORTB = &B01110011
Vaitas 1000
Loop
End
```

```

PORTB = &B00111000
Waitas 1000
PORTB = &B01110011
Waitas 1000
Loop
End

```

Z kodu powyższego programu można wywnioskować że aby zapalić dany znak lub cyfrę na wyświetlaczu 7-segmentowym należy na każdy z segmentów wyświetlacza dać odpowiednio 1 lub 0 co spowoduje że dany segment się zapali lub nie.

Program 2

Program 2 działa dosyć podobnie do programu 1. Wyświetla on liczby od 0 do 9 a następnie przez kilka iteracji pętli wyświetla literę E. Kod programu widać poniżej.

```

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

Config PORTB = Output

Declare Sub Wysw7seg(byval C As Byte)

Dim I As Byte
Dim Kod_cyf As Byte

Kropka Alias PORTB.7

Do
Toggle Kropka
For I = 0 To 15
Call Wysw7seg(i)
Waitas 500
Next I
Loop
End

Sub Wysw7seg(byval C As Byte)
If C < 10 Then
Kod_cyf = Lookup(C, Kody7seg)
Kod_cyf.7 = Kropka
PORTB = Kod_cyf
Else
PORTD = PORTB And &B10000000
PORTB = &B01111001
End If
End Sub

Kody7seg:
Data &B00111111, &B00000110, &B01011011, &B01001111, &B01100110,
Data &B01101101, &B01111101, &B00000111, &B01111111, &B01101111, &B01111001

```

Program działa w taki sposób, że w pętli wywoływana jest funkcja Wysw7seg, która przez pierwsze 10 kroków pętli wyświetla cyfry od 0 do 9, a następnie aż do zakończenia pętli wyświetla literę E. Program powtarza swoje działania, dzięki instrukcji Do Loop.

Program 3

Program 3 wykorzystywał wszystkie 4 wyświetlacze 7-segmentowe a jego zadaniem było wyświetlanie cyfr od 9 do 0, a następnie przejście do następnego wyświetlacza i wykonanie tej samej czynności. Można zatem powiedzieć że wyświetlacze były w pewien sposób synchronizowane, gdyż ich stan zależał od stanu wyświetlacza poprzedniego. Kod programu widać poniżej.

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

Config PORTD = Output
Config PINB.0 = Output
Config PINB.1 = Output
Config PINB.2 = Output
Config PINB.3 = Output
Config PINC.0 = Output
Config Timer0 = Timer, Prescale = 256

Declare Sub Pobr_znaku(cyfra As Byte)
On Timer0 Mult_wysw

Dim A As Byte, B As Byte, C As Byte, D As Byte
Dim Nr_wysw As Byte
Dim I As Byte
W1 Alias PORTB.0
W2 Alias PORTB.1
W3 Alias PORTB.2
W4 Alias PORTB.3

Enable Interrupts
Enable Timer0
Load Timer0, 125
Set PORTC.0
Do
For I = 9 To 0 Step -1
A=I
Wait 1
Next I
For I = 9 To 0 Step -1
B=I
Wait 1
Next I
For I = 9 To 0 Step -1
C=I
Wait 1
Next I
For I = 9 To 0 Step -1
D=I
Wait 1
Next I
Wait 1
A = 9 : B=9 : C=9 : D=9
Reset PORTC.0
A=10
B=11
C=12
D=13
Wait 10
Set PORTC.0
Loop
End

Sub Pobr_znaku(cyfra As Byte)
If Cyfra < 1 Then
PORTD = 0
Else
PORTD = Lookup(cyfra, Kody7seg)
End If
End Sub
```

```

Mult_wysw:
Load Timer0, 125
Set W1
Set W2
Set W3
Set W4
Select Case Nr_wysw

Case 0:
Call Pobr_znaku(a)
Reset W1
Case 1:
Call Pobr_znaku(b)
Reset W2
Case 2:
Call Pobr_znaku(c)
Reset W3
Case 3:
Call Pobr_znaku(d)
Reset W4
End Select
Incr Nr_wysw

If Nr_wysw = 4 Then
Nr_wysw = 0
End If
Return

Kody7seg:
Data &B00111111, &B00000110, &B01011011, &B01001111, &B01100110,
Data &B01101101, &B01111101, &B00000111, &B01111111, &B01101111, &B01110110, &B01111001, &B00111000, &B01110011

```

Program działa w taki sposób, że wyświetla na każdym z wyświetlaczy liczby od 9 do 0. Gdy wyświetlana wartość będzie w kształcie cyfry 0, wówczas program przechodzi do następnego wyświetlacza, ten zaś wykonuje tą samą czynność. Cała sytuacja trwa, aż na ostatnim wyświetlaczu zostanie wyświetlone 0. W ten sposób otrzymujemy pewną synchronizację wyświetlaczy. Wszystkie wymienione powyżej instrukcje są zapętlone dzięki Do Loop.

3. Wnioski

Program 1

Po przeanalizowaniu kodu programu 1 możemy wywnioskować że aby zapalić dany znak lub cyfrę na wyświetlaczu 7-segmentowym należy na każdy z segmentów wyświetlacza dać odpowiednio 1 lub 0 co spowoduje że dany segment się zapali lub nie.

Program 2

Program 2 działa w taki sposób, że w pętli wywoływana jest funkcja Wysw7seg, która przez pierwsze dziesięć kroków wyświetla cyfry od 0 do 9, a potem aż do zakończenia pętli wyświetla literę E.

Program 3

Program 3 wyświetla na każdym z wyświetlaczy liczby w kolejności od 9 do 0. Gdy wyświetlana wartość będzie w kształcie cyfry 0, wówczas program przechodzi do następnego wyświetlacza, natomiast kolejny wyświetlacz wykonuje tą samą czynność. Cała sytuacja trwa, aż na ostatnim wyświetlaczu zostanie wyświetlone 0. W ten sposób otrzymujemy pewną synchronizację wyświetlaczy.