



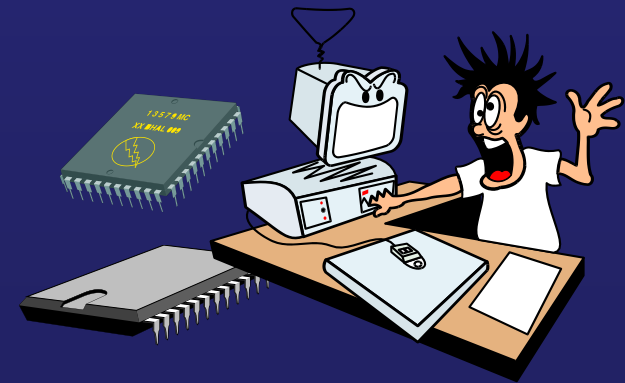
WYDZIAŁ FIZYKI
i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki



Systemy wbudowane



Witold Kozłowski



<https://std2.phys.uni.lodz.pl/mikroprocesory/>

Systemy wbudowane

Kierunek: Informatyka
PRACOWNIA DYDAKTYCZNA

Uwaga !!!

Proszę o wyłączenie
telefonów komórkowych

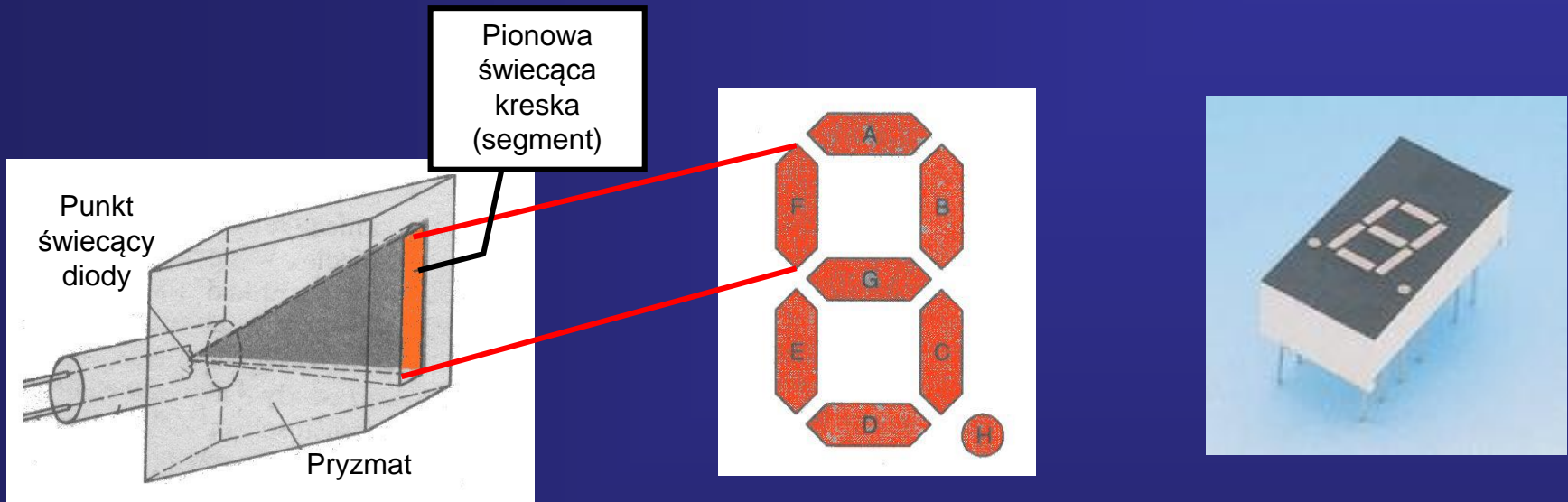
na wykładzie i laboratorium

Wykład 4.

**Sterowanie statyczne i multipleksowe
wyświetlaczami 7 – segmentowymi
LED**

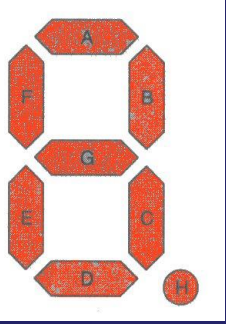
7- segmentowy wyświetlacz LED

Light Emitting Diode



Najprostszymi elementami sygnalizującymi stan mikrokontrolera są diody LED, ale do wyświetlenia cyfr czy znaków alfanumerycznych stosowane są różnego rodzaju wyświetlacze.

Wyświetlacz LED zbudowany jest z diod LED, które tworzą jego segmenty.



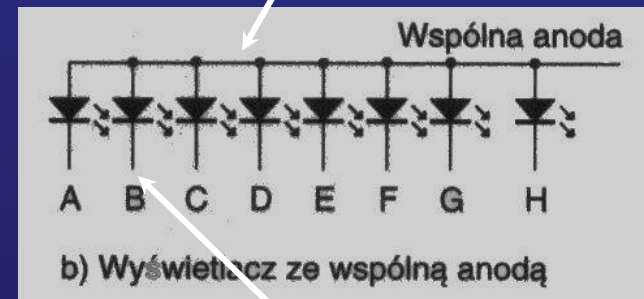
7-segmentowy wyświetlacz LED

Wspólna katoda

Wspólna anoda

0V

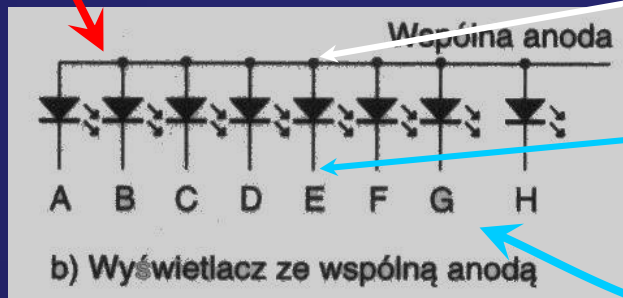
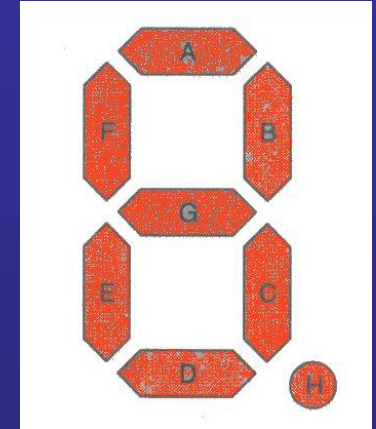
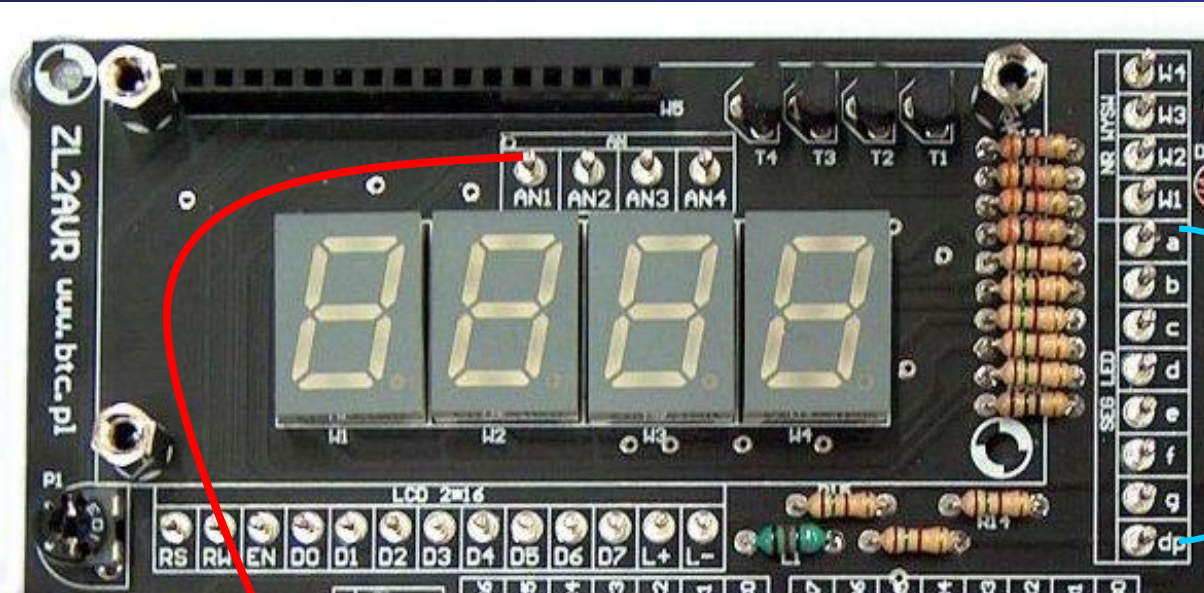
Okolo
+1.5V



Okolo
+1.5V

0V

Zestaw AL2AVR wyposażony jest w 4 wyświetlacze LED o wspólnej anodzie



Okolo
+1.5V

0V

Zestaw AL2AVR wyposażony jest w 4 wyświetlacze LED o wspólnej anodzie



Sterowanie anodami wyświetlaczy LED odbywa się poprzez tranzystory

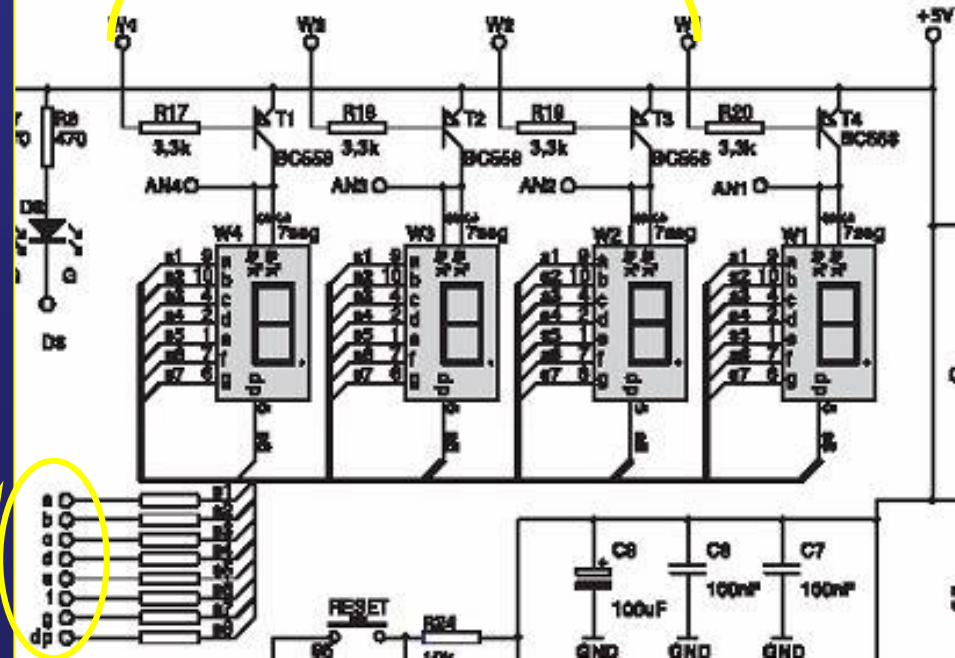
T1 – T4

Podanie „0” (potencjał 0V) na wejście W1-W4 powoduje załączenie poszczególnych wyświetlaczy

Wszystkie katody poszczególnych segmentów wyświetlaczy są połączone ze sobą.

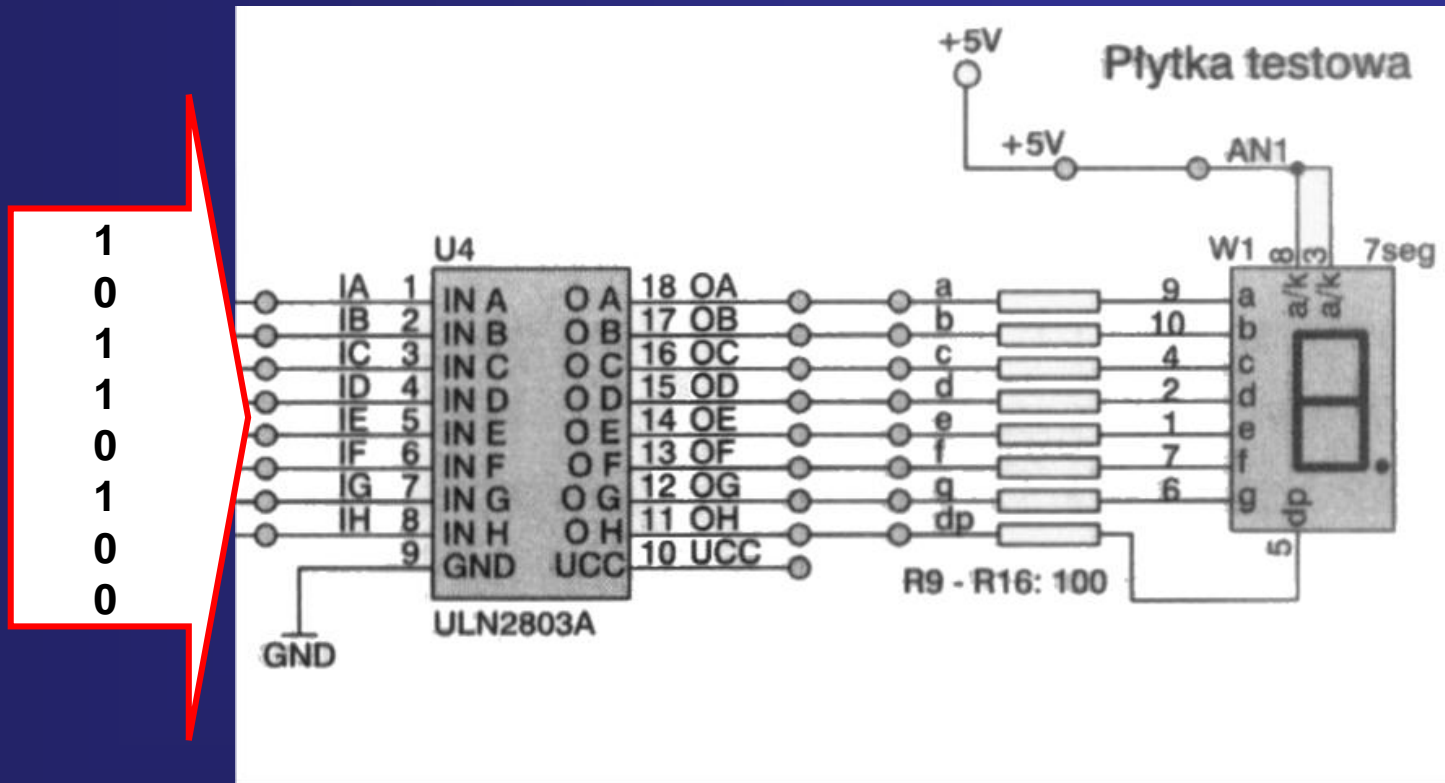
rezystory R1 – R8 ograniczają prąd.

Podanie „0” (potencjał 0V) na wejście a....g powoduje świecenie poszczególnych segmentów



Statyczne sterowanie

7 - segmentowego wyświetlacza LED ze wspólną anodą poprzez układ mocy ULN2803A



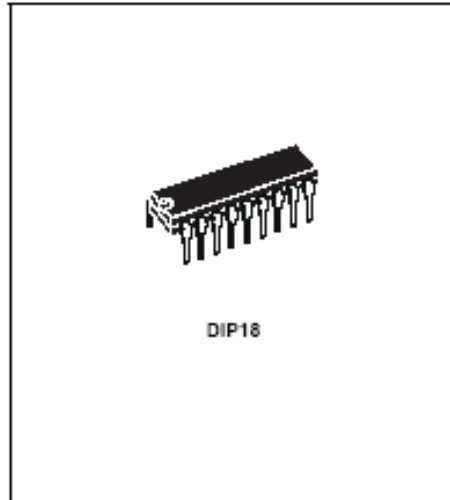
Układ mocy ULN2803A



ULN2801A
ULN2802A - ULN2803A
ULN2804A - ULN2805A

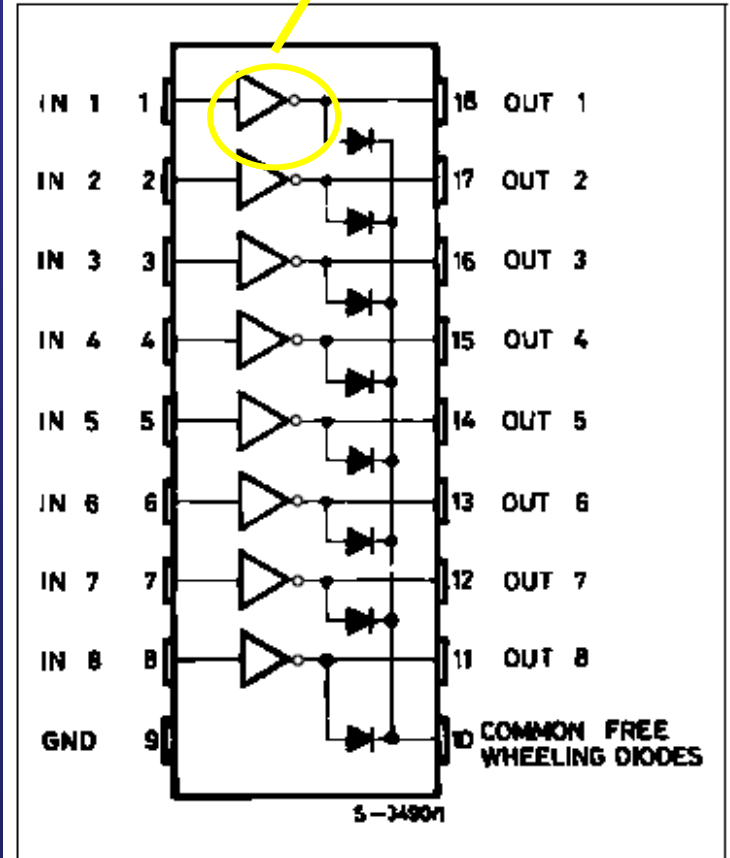
EIGHT DARLINGTON ARRAYS

- EIGHT DARLINGTONS WITH COMMON EMITTERS
- OUTPUT CURRENT TO 500 mA
- OUTPUT VOLTAGE TO 50 V
- INTEGRAL SUPPRESSION DIODES
- VERSIONS FOR ALL POPULAR LOGIC FAMILIES
- OUTPUT CAN BE PARALLELED
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY BOARD LAYOUT



NOT
„0” ⇒ „1”
„1” ⇒ „0”

PIN CONNECTION (top view)



Statyczne sterowanie

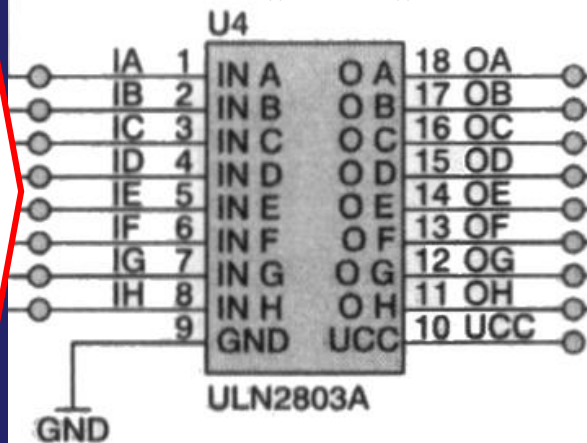
7 - segmentowego wyświetlacza LED ze wspólna anodą
poprzez układ mocy ULN2803A

NOT

„0” ⇒ „1”

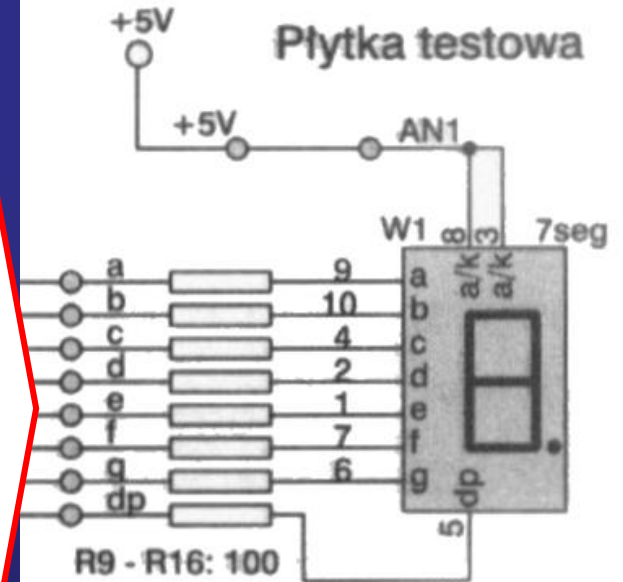
„1” ⇒ „0”

1
1
1
1
0
0
0
0

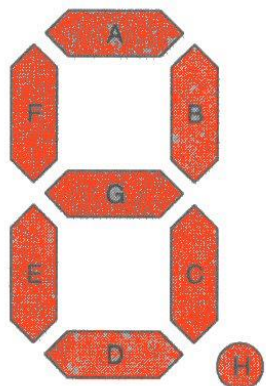


0
0
0
0
0
1
1
1
1
1

Płytki testowa



Słowa kodów 7 - segmentowego wyświetlacza LED ze wspólna anoda sterowanego poprzez układ ULN2803N



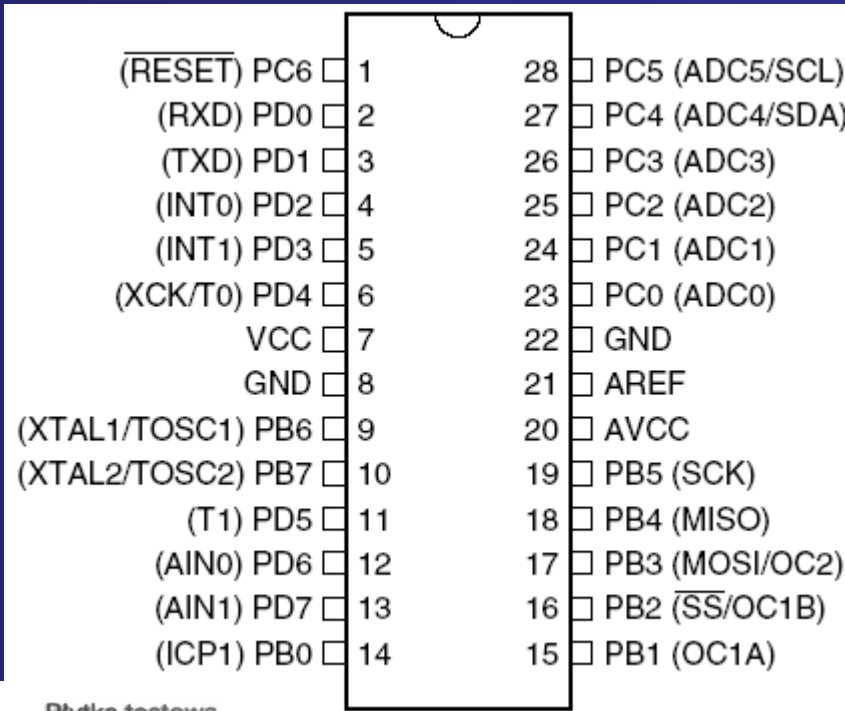
Wyświetlana cyfra	Kody cyfr - sterujące 7- segmentowym wyświetlaczem LED o wspólnej anodzie przez układ ULN2803A							
	dp	g	f	e	d	c	b	a
0	0	0	1	1	1	1	1	1
1	0	0	0	0	0	1	1	0
2	0	1	0	1	1	0	1	1
3	0	1	0	0	1	1	1	1
4	0	1	1	0	0	1	1	0
5	0	1	1	0	1	1	0	1
6	0	1	1	1	1	1	0	1
7	0	0	0	0	0	1	1	1
8	0	1	1	1	1	1	1	1
9	0	1	1	0	1	1	1	1

W tablicy przedstawiono słowa kodowe powodujące wyświetlenie na wyświetlaczu LED cyfry 0...9

Statyczne sterowanie 7 - segmentowego wyświetlacza LED

Port D	Wyk.
PD7	6 wyj./wej. do wykorzystania
PD6	
PD5	
PD4	
PD3	
PD2	
PD1	TXD
PD0	RXD

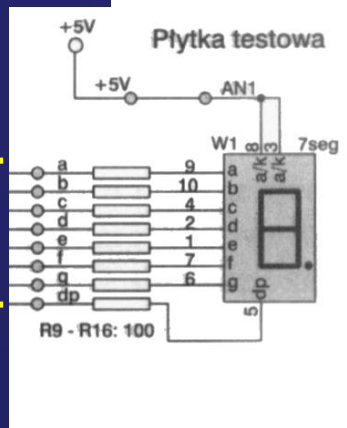
Wykorzystane do symulacji sprzętowej transmisja RS 232



Port B	Wyk.
PB7	XTAL1
PB6	XTAL2
PB5	6 wyj./wej. do wykorzystania
PB4	
PB3	
PB2	
PB1	
PB0	

Oscylator rezystor kwarcowy

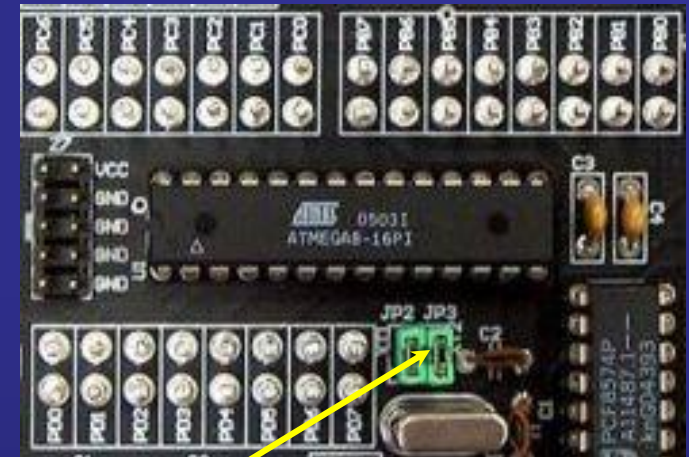
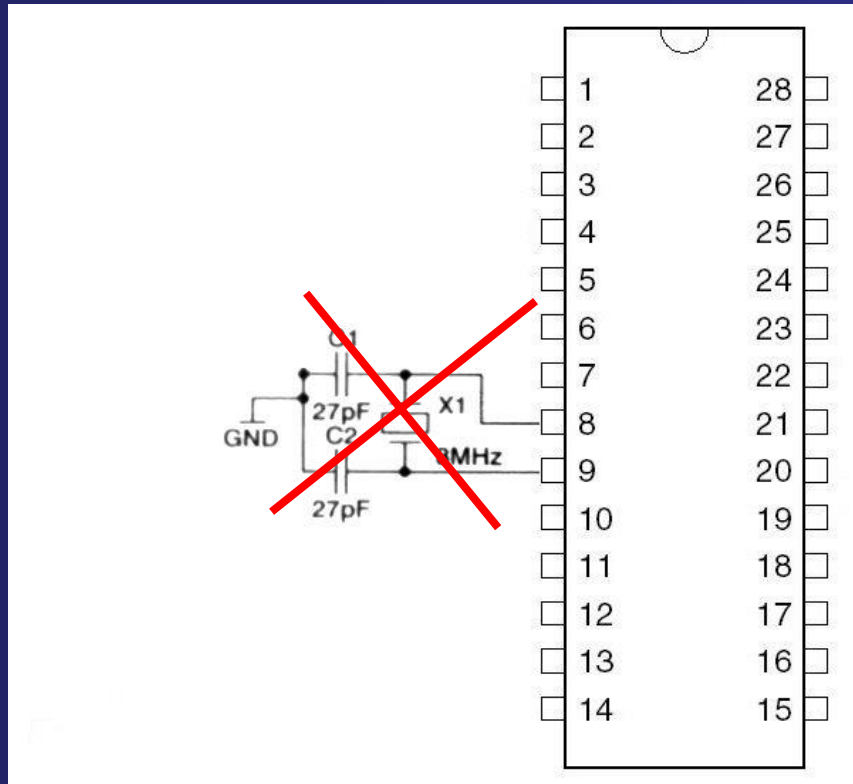
Port C	Wyk.
PC7	-
PC6	RESET
PC5	6 wyj./wej. do wykorzystania
PC4	
PC3	
PC2	
PC1	
PC0	



Aby sterować wyświetlaczem LED

potrzebujemy 8 wolnych linii portu

Statyczne sterowanie 7 - segmentowego wyświetlacza LED



Usunąć zwory
JP2, JP3

Jeśli nie jest potrzebne dokładne odmierzanie czasu, można skorzystać z wewnętrznego oscylatora RC w jaki wyposażono jest nasz mikrokontroler. Dzięki czemu zyskujemy dodatkowe dwa wyjścia portu B

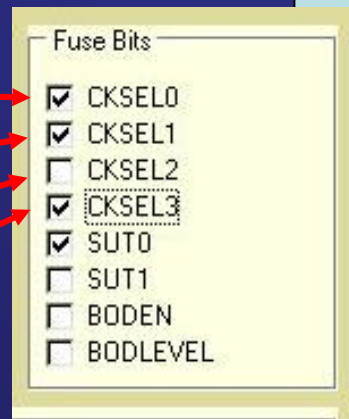
Konfiguracja *Fuse bits* w celu uaktywnienia wewnętrznego oscylatora RC

Table 9. Internal Calibrated RC Oscillator Operating Modes

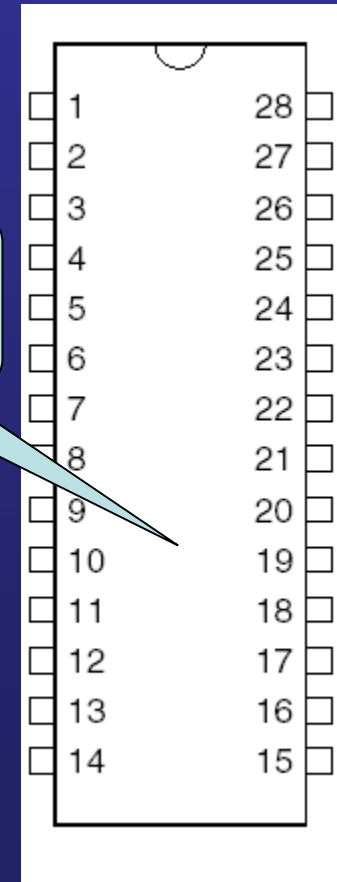
CKSEL3..0	Nominal Frequency (MHz)
0001 ⁽¹⁾	1.0
0010	2.0
0011	4.0
0100	8.0

Note: 1. The device is shipped with this option selected.

Nr bitu	Nazwa Fuse Bitu	Wartość Fuse Bitu
0	CKSEL 0	0
1	CKSEL 1	0
2	CKSEL 2	1
3	CKSEL 3	0
4	SUTO 0	0
5	SUTO 1	1
6	BODEN	1
7	BODLEVEL	1

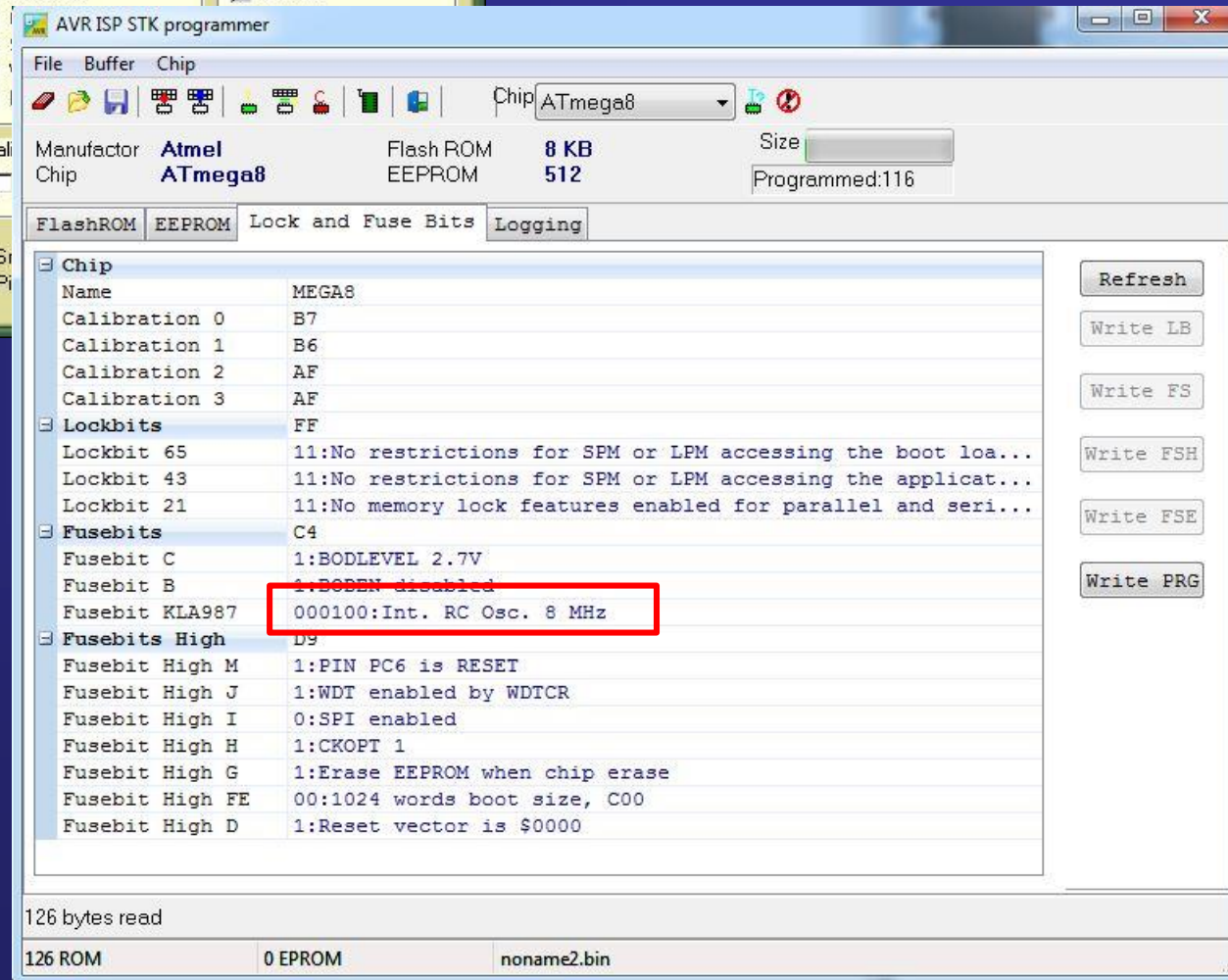
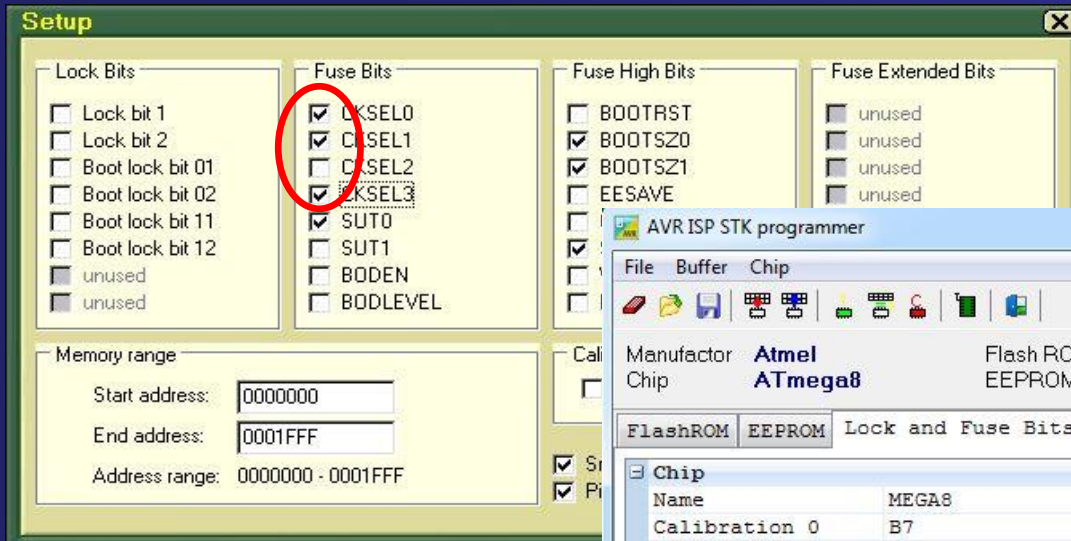


Oscillator RC inside

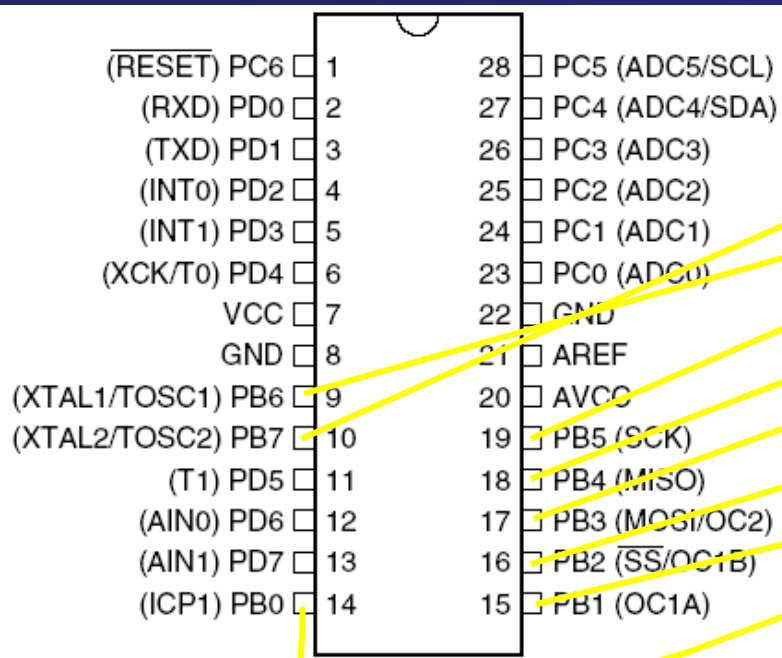


Programowanie

Odczytanie zawartości konfiguracji bitów Fuse

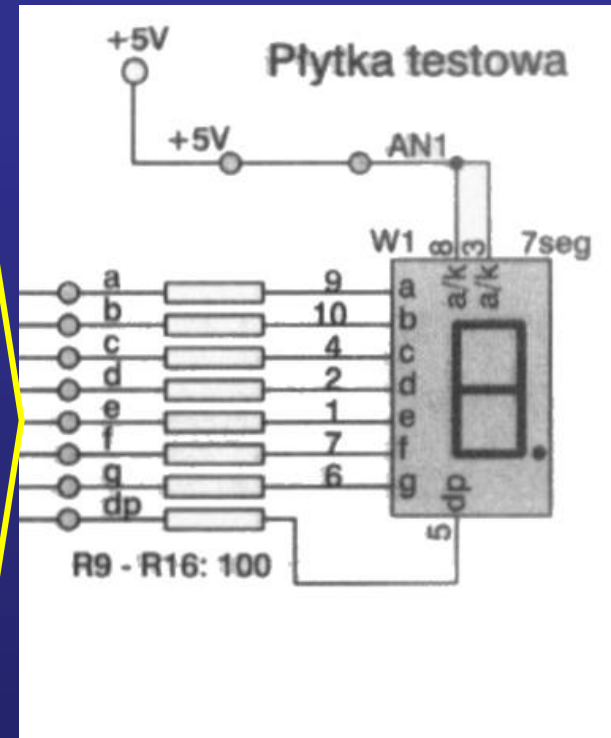


Statyczne sterowanie 7 - segmentowego wyświetlacza LED



Port B	Wyk.
PB7	dp
PB6	g
PB5	f
PB4	e
PB3	d
PB2	c
PB1	b
PB0	a

8 wyj./wej. do wykorzystania



Wszystkie wyjścia portu B mogą zostać podłączone do wyświetlacza LED

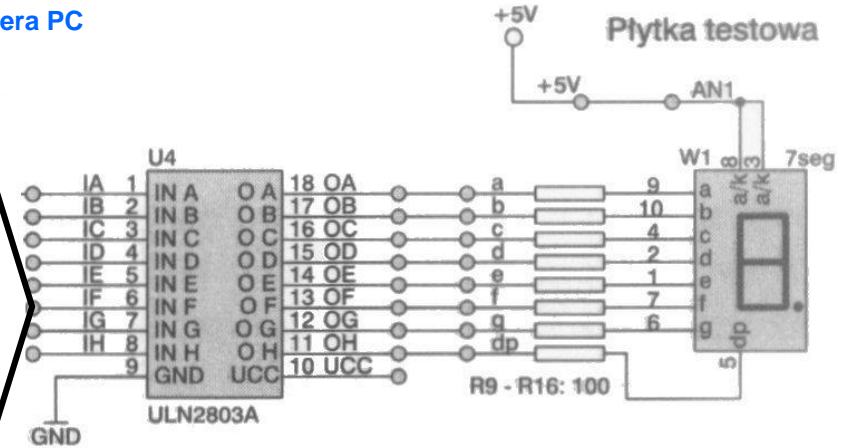
Program 6

Statyczne sterowanie 7 - segmentowego
wyświetlacza LED

Statyczne sterowanie 7 - segmentowego wyświetlacza LED

RS232 do komputera PC

Port
PB0...PB7
Do
Wyświetlacza
LED



```

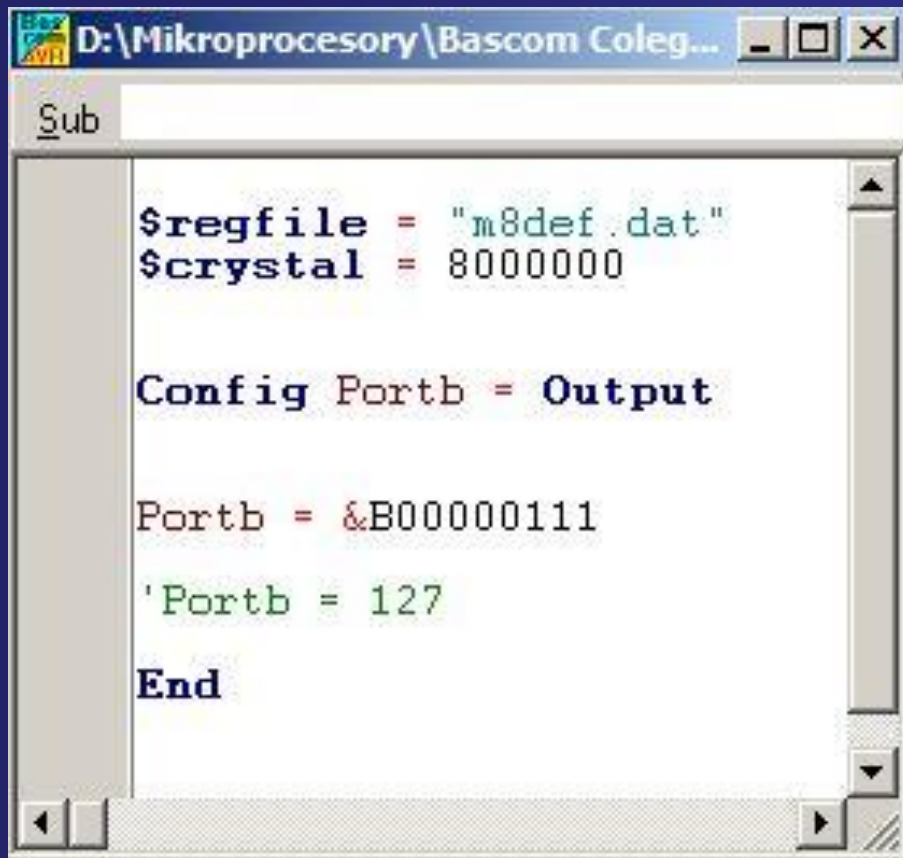
C:\Bascom\College\basAVR_listingi\02\zajecia\basmon bez opisu.bas
Sub
$regfile = "at8644.dat"
$crystal = 8000000
$baud = 19200
[variables]
Dim Kik As Byte
Dim Adr As Word
Dim Adr1 As Byte , Adr2 As Byte
Dim V1 As Byte
[main program]
Print "BASCOM Version 1.03"
Do
  Kik = Inkey()
  If Kik = "T" Then
    Print Chr(13)
  Elseif Kik = "U" Then
    Adr = Waitkey()
    V1 = Waitkey()
    Out Adr, V1
    Print Chr(13)
  Elseif Kik = "R" Then
    Adr = Waitkey()
    V1 = Inp(adr)
    Print Chr(v1)
  Elseif Kik = "0" Then
    Adr1 = Waitkey()
    Adr2 = Waitkey()
    V1 = Waitkey()
    Adr = Adr1 * 256
    Adr = Adr + Adr2
    Out Adr, V1
    Print Chr(13)
  Elseif Kik = "?" Then
    Print "?"
  End If
Loop
  
```

Wpisanie programu monitora do pamięci Flash ROM

Port B	Wyk.
PB7	dp
PB6	g
PB5	f
PB4	e
PB3	d
PB2	c
PB1	b
PB0	a

Program 6

Statyczne sterowanie 7 - segmentowego wyświetlacza LED



```
Sub  
  
$regfile = "m8def.dat"  
$crystal = 8000000  
  
Config Portb = Output  
  
Portb = &B00000111  
  
'Portb = 127  
  
End
```

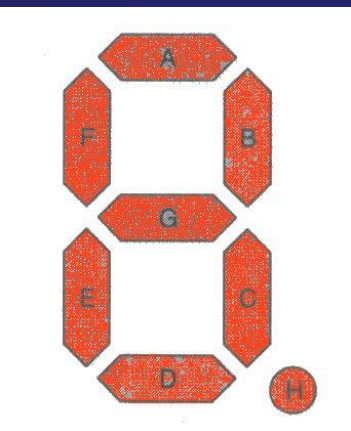
Program 6

Statyczne sterowanie 7 - segmentowego wyświetlacza LED

Zadania do wykonania na Ćwiczeniach:

Sprawdzenie tabeli kodów cyfr dla wyświetlacza LED

7- segmentowego - sterowanego układem ULN2803N



Wyświetlana cyfra	Kody cyfr - sterujące 7- segmentowym wyświetlaczem LED o wspólnej anodzie przez układ ULN2803A							
	dp	g	f	e	d	c	b	a
0	0	0	1	1	1	1	1	1
1	0	0	0	0	0	1	1	0
2	0	1	0	1	1	0	1	1
3	0	1	0	0	1	1	1	1
4	0	1	1	0	0	1	1	0
5	0	1	1	0	1	1	0	1
6	0	1	1	1	1	1	0	1
7	0	0	0	0	0	1	1	1
8	0	1	1	1	1	1	1	1
9	0	1	1	0	1	1	1	1

Statyczne sterowanie 7 - segmentowego wyświetlacza LED

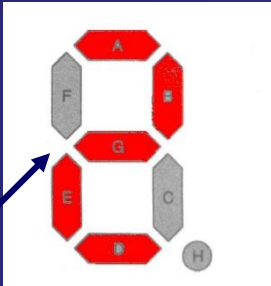
```

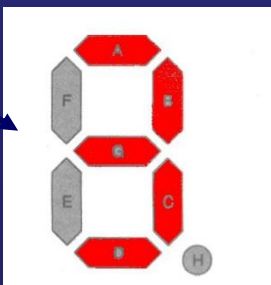
D:\Mikroprocesory\Bascom Coleg...
Sub
$regfile = "m8def.dat"
$crystal = 8000000

Config Portb = Output

Portb = &B01011011
Portb = 79

End
    
```

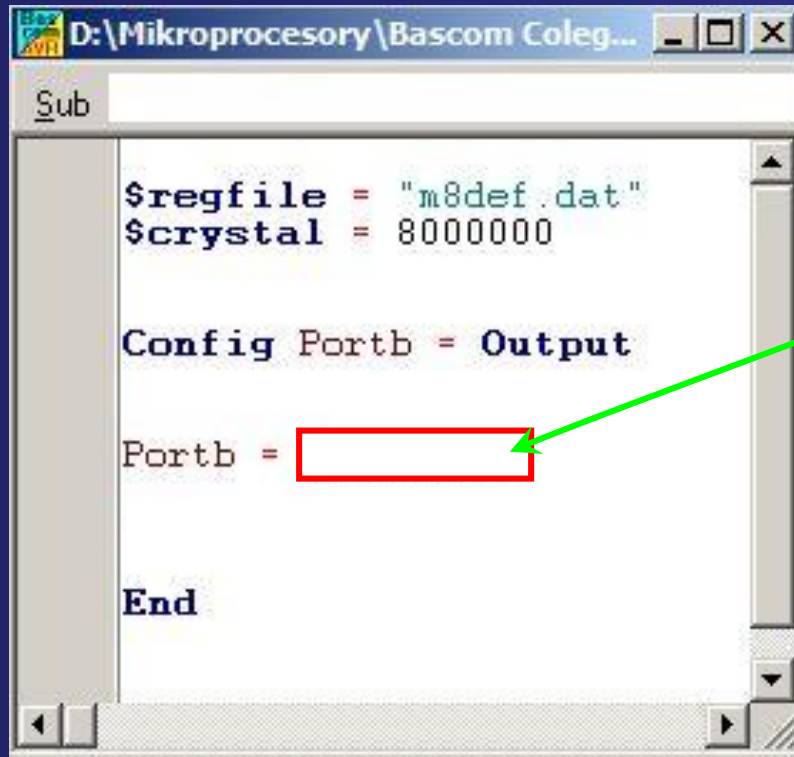
Wyświetlana cyfra	Port B	segmenty	Wpisanie do Portu B	Wartość binarna
	PB7	dp	0	01011011
	PB6	g	1	
	PB5	f	0	
	PB4	e	1	
	PB3	d	1	
	PB2	c	0	
	PB1	b	1	
	PB0	a	1	

Wyświetlana cyfra	Port B	segmenty	Wpisanie do Portu B	Wartość dziesiętna
	PB7	dp	0	79
	PB6	g	1	
	PB5	f	0	
	PB4	e	0	
	PB3	d	1	
	PB2	c	1	
	PB1	b	1	
	PB0	a	1	

Zadania do wykonania na Ćwiczeniach:

Ponieważ wyświetlacz posiada 8 wejść
sterowany jest słowem 8-bitowym (1bajt)

00000000....11111111 \Rightarrow 0...255



```
Sub  
$regfile = "m8def.dat"  
$crystal = 8000000  
  
Config Portb = Output  
  
Portb =   
  
End
```

Mozemy wpisać dowolną wartość z zakresu 0...255

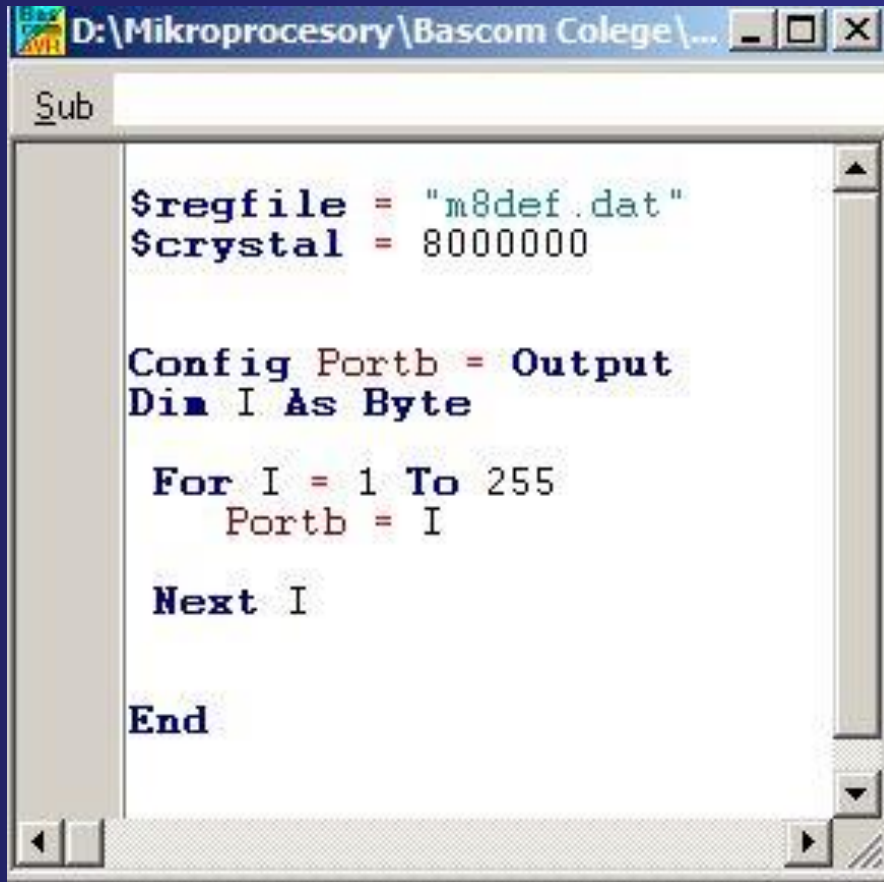
Należy wyznaczyć (liczby dziesiętne z zakresu 0...255), którym odpowiadają cyfry wyświetlacza 0..9

Program 6a

Statyczne sterowanie 7 - segmentowego
wyświetlacza LED

Program 6a

Statyczne sterowanie 7 - segmentowego wyświetlacza LED



```
Sub  
$regfile = "m8def.dat"  
$crystal = 8000000  
  
Config Portb = Output  
Dim I As Byte  
  
For I = 1 To 255  
    Portb = I  
  
Next I  
  
End
```

Program 6b

Statyczne sterowanie 7 - segmentowego
wyświetlacza LED

Wykorzystanie procedury wyświetlania tylko cyfr
od 0 do 9

```

D:\Mikroprocesory\Bascom College\basAVR_listingi\8_6_bez_opis_zmien_C.bas
Sub      Label
$sim
$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 9
        Call Wysw7seg(i)
        Waitms 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
    End If
End Sub

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

```

deklaracja procedury obsługi wyświetlacza LED

przypisanie bitowi 7 rejestru PORTB nazwy Kropka

wywołanie procedury Wysw7seg

definicja procedury Wysw7seg

zapisz do zmiennej Kod_cyf kod znaku z tablicy stałych Kody7seg

zapis bitu 7 (sterującego kropką) do bitu 7 zmiennej Kod_cyf
zapobiegnie to zmianie stanu linii sterującej kropka

wartość zmiennej Kod_cyf wystawiana jest na port B

wygasza wyświetlaną cyfrę bez zmiany stanu kropki

tablica stałych Kody7seg
przechowująca kody
wyświetlanych cyfr Kody cyfr

0

1

2

3

4

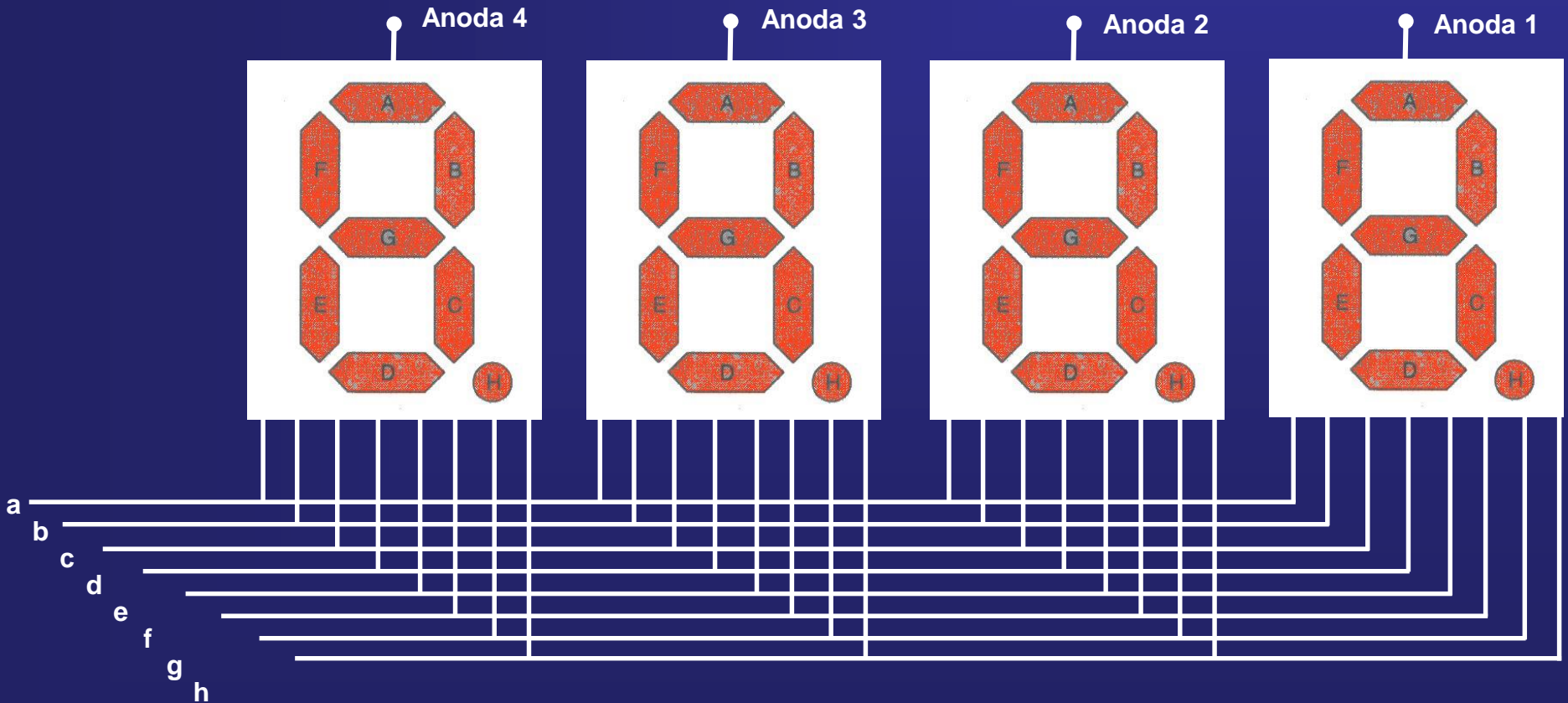
MULTIPLEKSOWANIE

Sterowanie MULTIPLEKSOWE
7 - segmentowymi wyświetlaczami LED

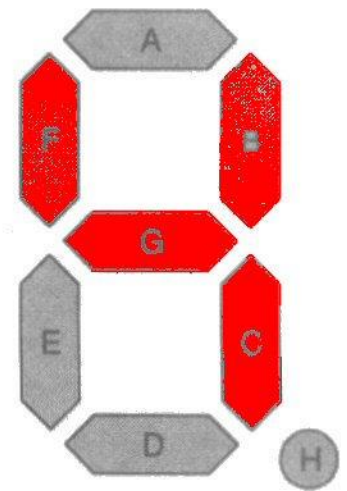
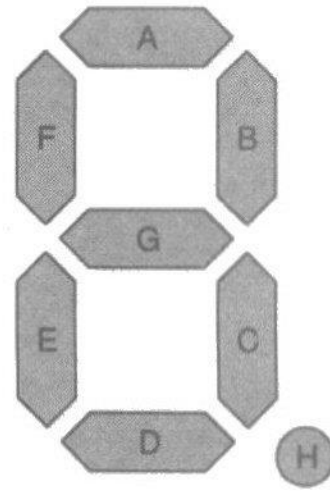
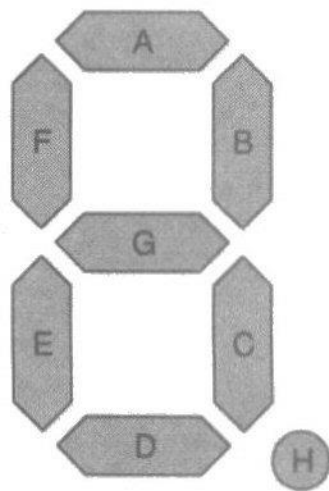
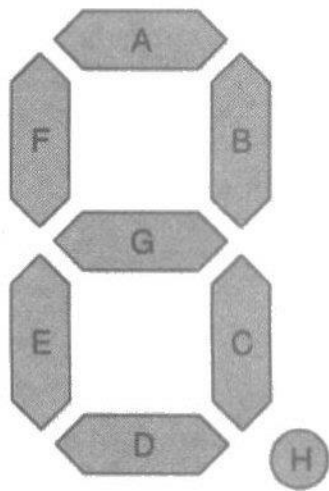
W większości przypadków jeden wyświetlacz LED nie wystarcza do prezentowania wyników !!!

Wykorzystanie sterowania multipleksowego minimalizuje liczbę potrzebnych do sterowania linii
do $8 + \text{liczba wyświetlaczy}$.

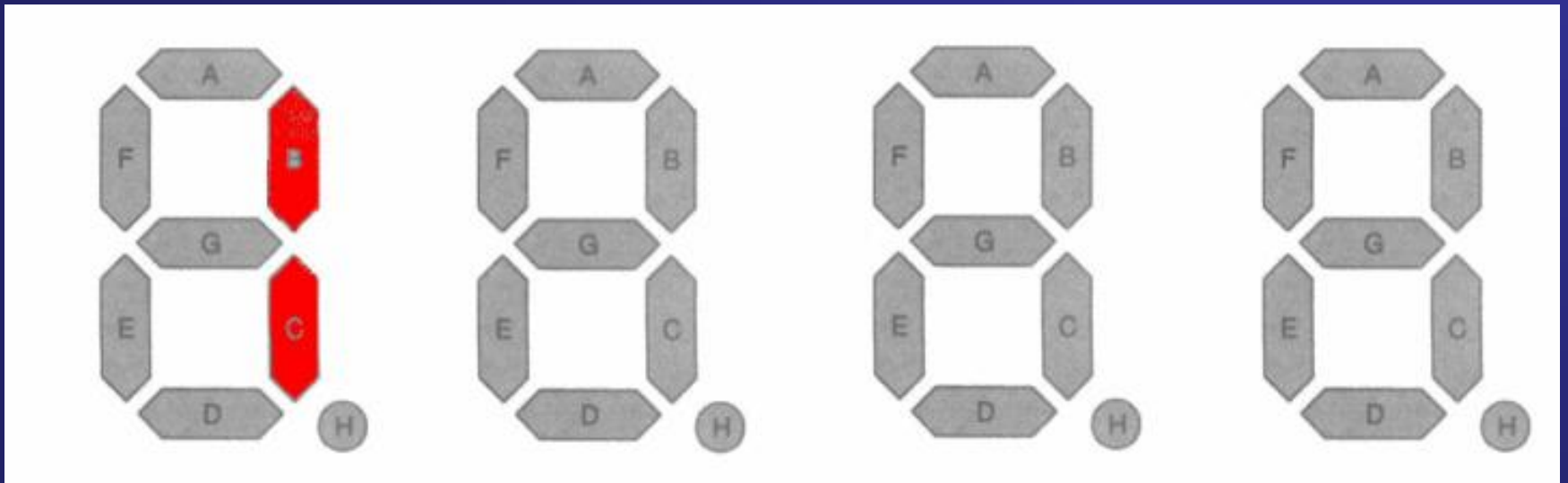
W naszym przypadku będziemy sterować 4 wyświetlaczami a więc $8 + 4$ daje 12, czyli będziemy potrzebować 12 linii mikroprocesora a nie 32 jak przy sterowaniu statycznym.



Sterowanie MULTIPLEKSOWE 7 - segmentowymi wyświetlaczami LED

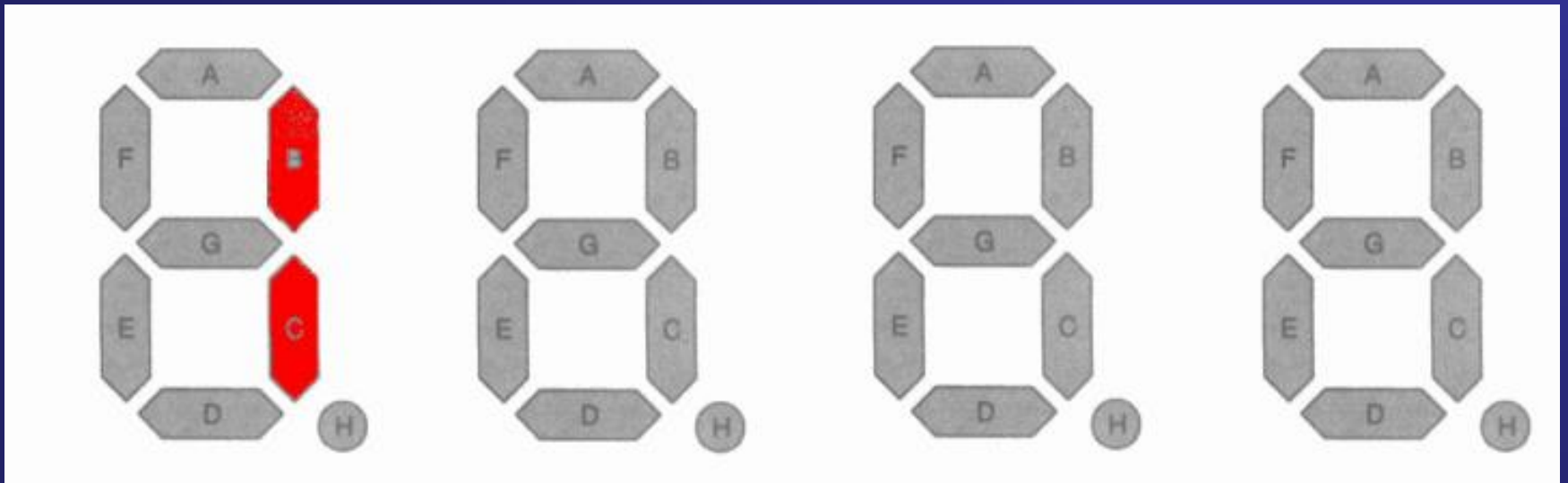


Sterowanie MULTIPLEKSOWE 7 - segmentowymi wyświetlaczami LED



Okolo 1Hz

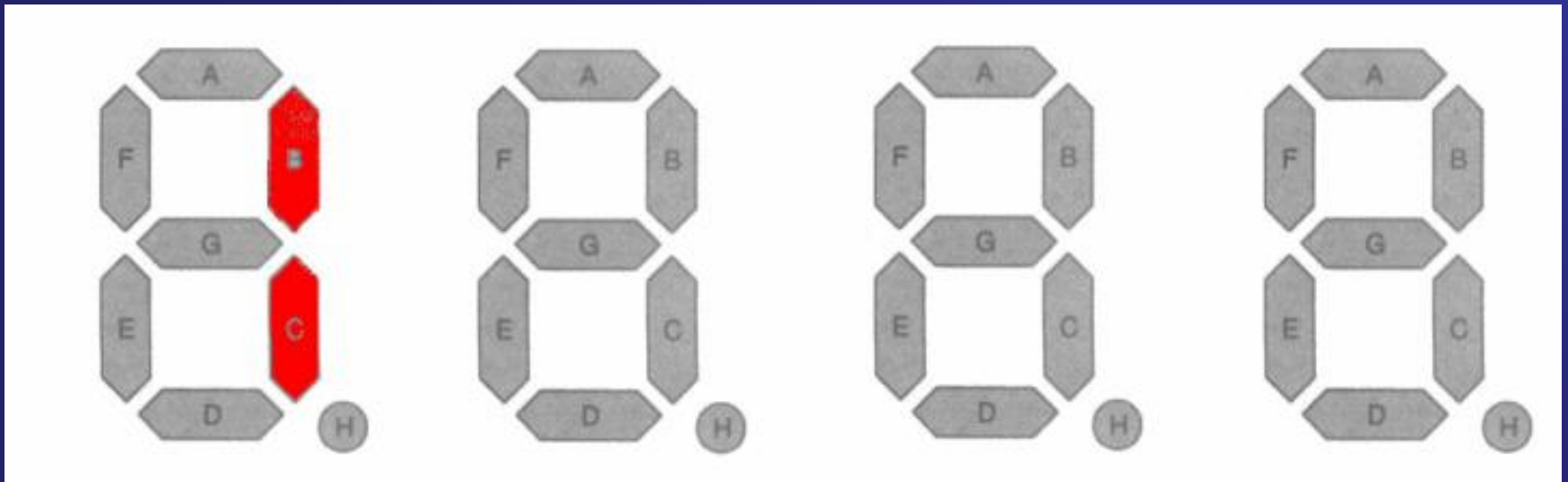
Sterowanie MULTIPLEKSOWE 7 - segmentowymi wyświetlaczami LED



Okolo 3Hz

Sterowanie MULTIPLEKSOWE 7 - segmentowymi wyświetlaczami LED

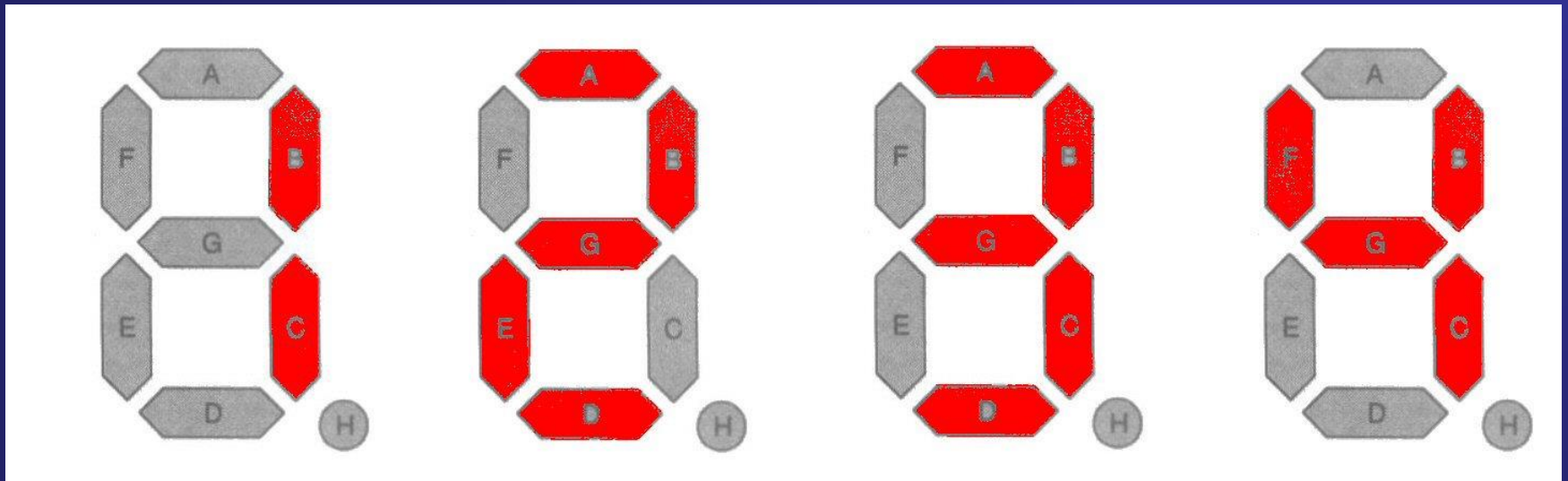
Około 8Hz



Aby migotanie wyświetlanych cyfr nie było zauważalne dla oka ludzkiego, częstotliwość multipleksowania (załączania) jednego wyświetlacza powinna wynosić co najmniej **25 Hz**.

Przy obsłudze 4 wyświetlaczy częstotliwość ich sterowania powinna wynosić co najmniej $4 * 25\text{Hz} = 100\text{Hz}$

Sterowanie MULTIPLEKSOWE 7 - segmentowymi wyświetlaczami LED



Częstotliwość sterowania większa od 25Hz dla jednego wyświetlacza

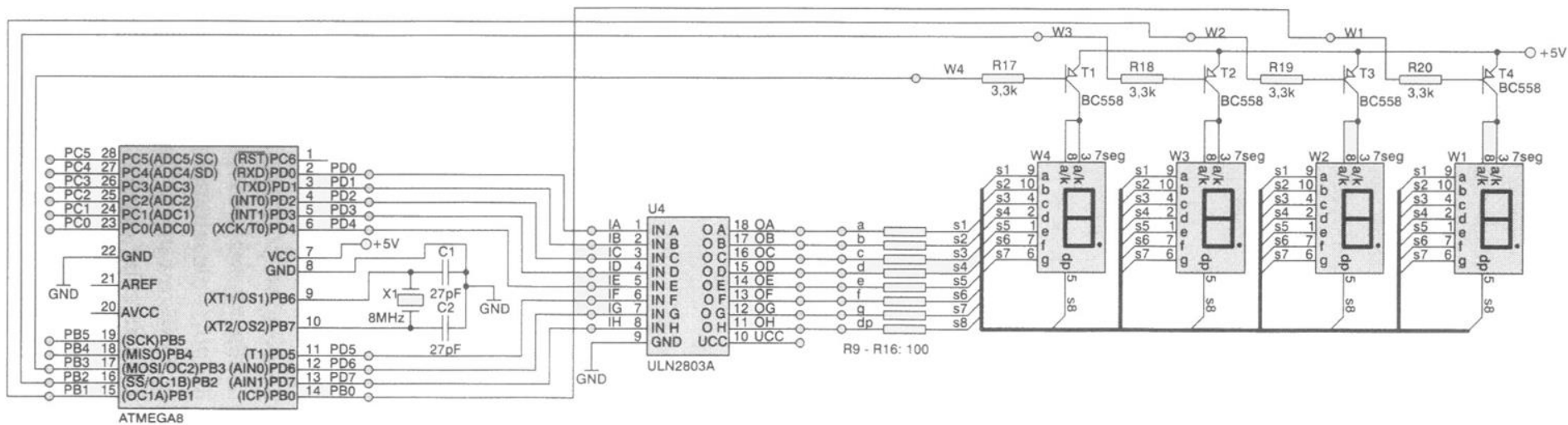
Program 7

Sterowanie
MULTIPLEKSOWE

7 - segmentowymi
wyświetlaczami LED

Schemat układu z multipleksowym sterowaniem

7 - segmentowych wyświetlaczy LED ze wspólną anodą poprzez układ mocy ULN2803A



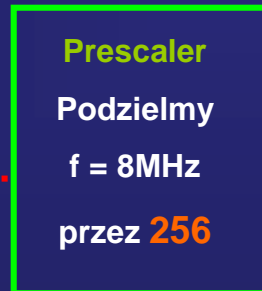
Wykorzystamy Timer0 - czasomierz

Do multipleksowego sterowania wyświetlaczami LED

Dla naszego oscylatora kwarcowego

$f = 8\text{MHz}$

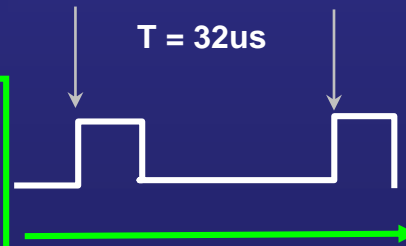
$T = 125\text{ns}$



Po podziale

$f = 31.25\text{kHz}$

$T = 32\mu\text{s}$



Uwaga:

należy ustawić licznik tak aby zliczył :

125 impulsów



Wygenerowanie przerwania po zliczeniu 125 impulsów

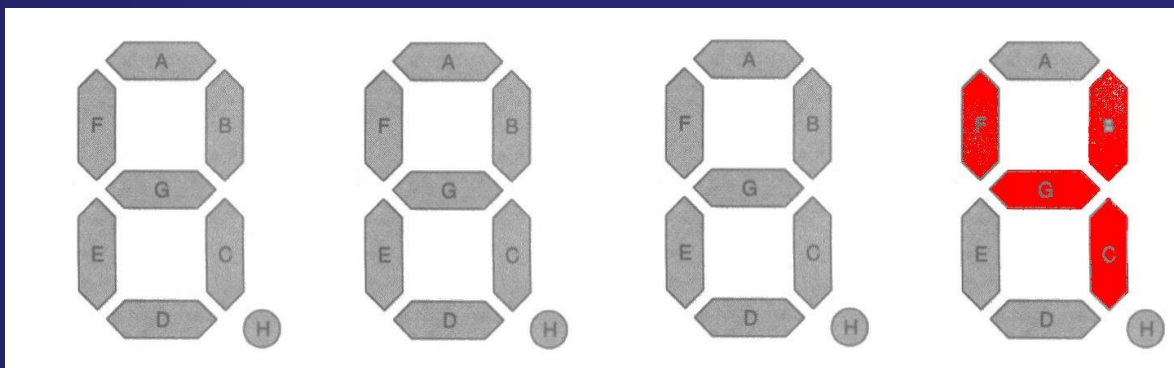
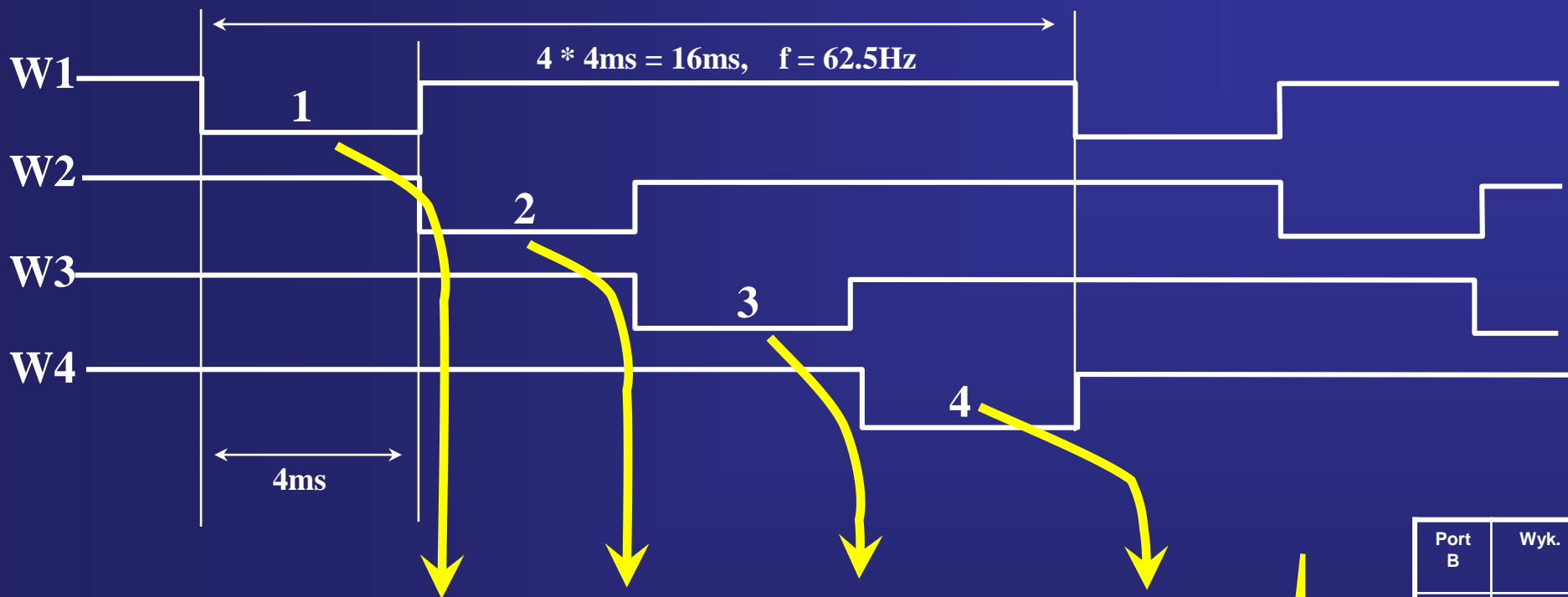


Przerwanie będzie generowane co

4ms

Program7

Sterowanie MULTIPLEKSOWE 7 - segmentowymi wyświetlaczami LED



Port B	Wyk.
PD7	dp
PD6	g
PD5	f
PD4	e
PD3	d
PD2	c
PD1	b
PD0	a

01001010

Program7

Sterowanie MULTIPLEKSOWE 7 - segmentowymi wyświetlaczami LED

```
D:\Mikroprocesory\Bascom Colege\basAVR_listingi\Dziala_zajecia\LED_8_...
Sub
Label

$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 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
```

Configuracja portu D i B jako wyjścia

Konfiguracje Timera0 jako czasomierza z podz. 256

Deklaracja procedury pobierającej kod cyfry do wyświetlania z tablicy

Przerwanie od przepełnienia Timera0, skok nastąpi do podprogramu Mult_wysw

Deklaracja zmiennych i aliansów

Odblokowanie przerwań globalne i od Timera0

Załadowanie do licznika wartości początkowej bo $256-125=131$

Program7

Sterowanie MULTIPLEKSOWE 7 - segmentowymi wyświetlaczami LED

```
D:\Mikroprocesory\Bascom Colege\basAVR_listingi\Dziala_zaj...
Sub                               Label
Do
For I = 1 To 10
  A = I
  Wait 1
Next I
For I = 1 To 10
  B = I
  Wait 1
Next I
For I = 1 To 10
  C = I
  Wait 1
Next I
For I = 1 To 10
  D = I
  Wait 1
Next I
Wait 5
A = 0 : B = 0 : C = 0 : D = 0
Loop
End
```

Początek nieskończonej pętli

Pętla wykonywana 10 razy, przypisania zmiennej A wartości I oraz wykonanie opóźnienia 1 s.

Wyzerowanie zmiennych A, B, C, D,

Program7

```
D:\Mikroprocesory\Bascom Colege\basAVR_listingi\Dziala_zajecia\LED_8_7bez_opisu
Sub
Label
Sub Pobr_znaku(cyfra As Byte)
  If Cyfra < 10 Then
    Portd = Lookup(cyfra , Kody7seg )
  Else
    Portd = 0
  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
```

definicja procedury Pobr_znaku, jeżeli wartość otrzymanej do wyświetlenia cyfry < od 10, to wpisanie do portu D wartość kodu z tablicy stałych Kody7seg w przeciwnym razie wpisanie do portu D wartości 0 która wygasza wyświetlacz

podprogramu obsługi przerwania od przepełnienia Timer0 w którym odbywa się multipleksowanie **wywoływane co 4ms**

w zależności od wartości Nr_wysw, wykonaj:

gdy Nr_wysw = 0, to wywołanie procedury pobierającej z tablicy kod znaku dla wartości A

zwiększenie o jeden wartości Nr_wysw

jeśli wartość Nr_wysw osiągnie 4, to zeruj wartość Nr_wysw

tablica stałych Kody7seg przechowująca kody wyświetlanych cyfr

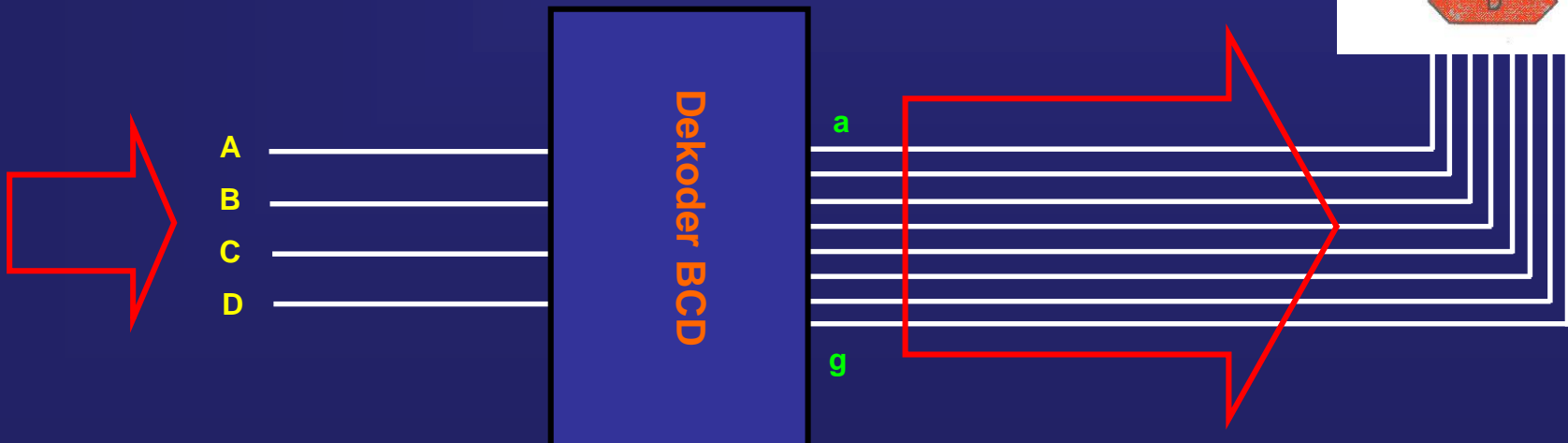
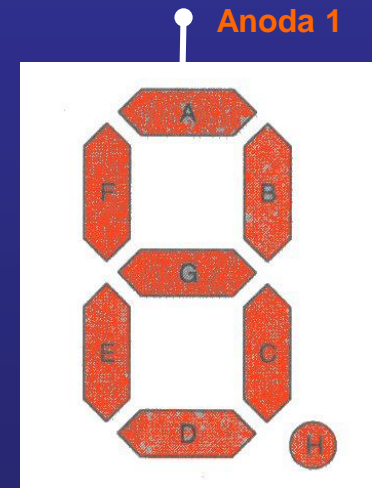
Program 8

Sterowanie multipleksowe

7 - segmentowych wyświetlaczy LED ze
wspólną anodą wykorzystując dekodery BCD
4543

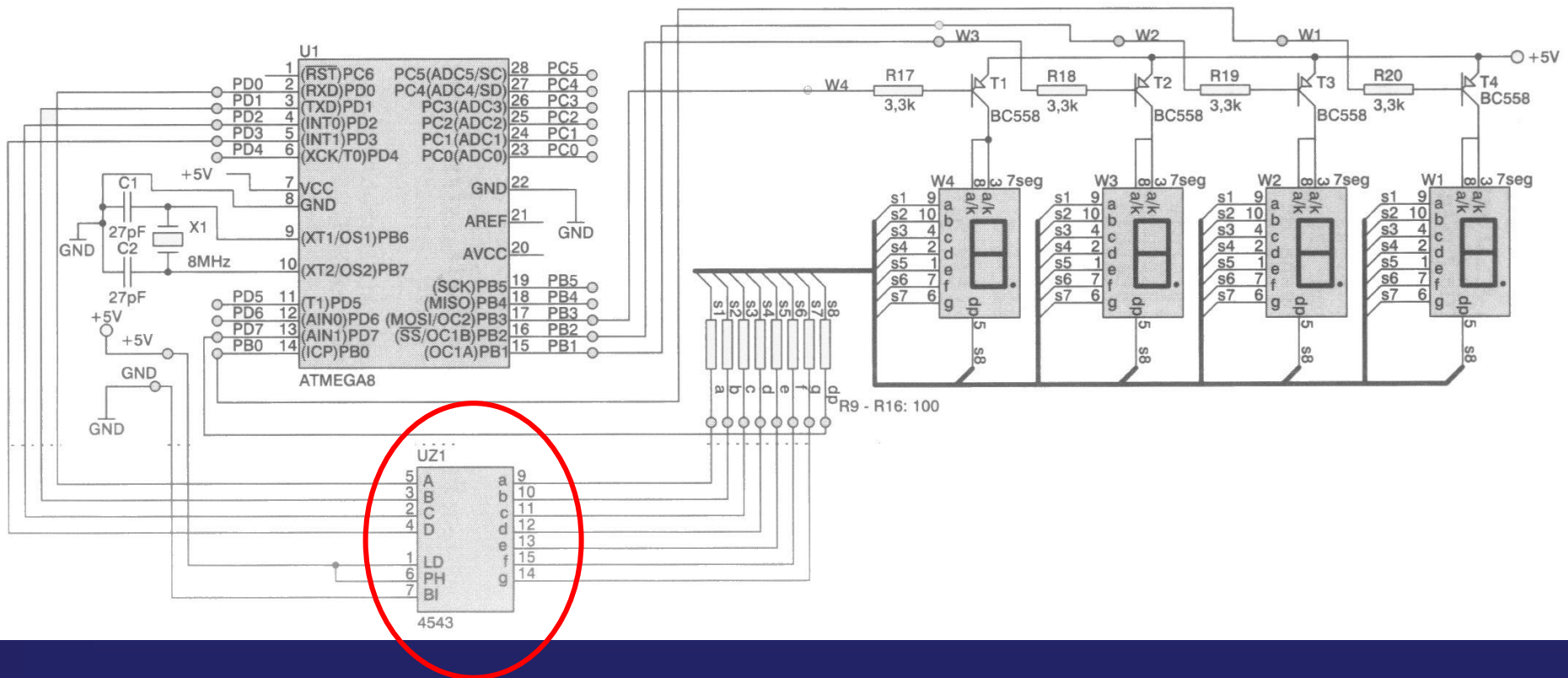
Dekoder kodu BCD na kod wyświetlacza 7 - segmentowego LED

Wyświetla na cyfra	Wejście dekodera BCD				Kody cyfr - 7- segmentowego wyświetlacza						
	D	C	B	A	g	f	e	d	c	b	a
0	0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0	1	1	0
2	0	0	1	0	1	0	1	1	0	1	1
3	0	0	1	1	1	0	0	1	1	1	1
4	0	1	0	0	1	1	0	0	1	1	0
5	0	1	0	1	1	1	0	1	1	0	1
6	0	1	1	0	1	1	1	1	1	0	1
7	0	1	1	1	0	0	0	0	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	0	1	1	1	1



Program 8

Schemat układu z multipleksowym sterowaniem 7 - segmentowych wyświetlaczy LED ze wspólną anodą wykorzystując dekodery BCD 4543



DATA SHEET

For a complete data sheet, please also download:

- The IC04 LOCMOS HE4000B Logic Family Specifications HEF, HEC
- The IC04 LOCMOS HE4000B Logic Package Outlines/Information HEF, HEC

HEF4543B

MSI

BCD to 7-segment
latch/decoder/driver

Product specification
File under Integrated Circuits, IC04

January 1995

Dekoder kodu BCD na kod wyświetlacz 7 - segmentowego LED



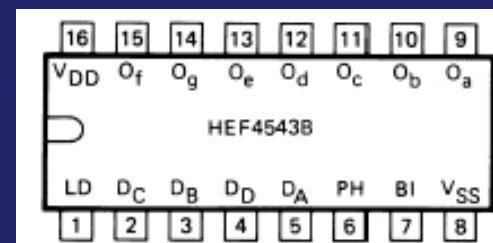
PHILIPS

HEF4543B

INPUTS							OUTPUTS							
LD	BI	PH ⁽⁴⁾	D _D	D _C	D _B	D _A	O _a	O _b	O _c	O _d	O _e	O _f	O _g	DISPLAY
X	H	L	X	X	X	X	L	L	L	L	L	L	L	blank
H	L	L	L	L	L	L	H	H	H	H	H	H	L	0
H	L	L	L	L	L	H	L	H	L	L	L	L	L	1
H	L	L	L	L	H	L	H	H	L	H	H	L	H	2
H	L	L	L	L	H	H	H	H	H	L	L	L	H	3
H	L	L	L	H	L	L	L	H	L	L	L	H	H	4
H	L	L	L	H	L	H	H	L	H	L	L	H	H	5
H	L	L	L	H	H	L	H	L	H	H	H	H	H	6
H	L	L	L	H	H	H	H	H	L	L	L	L	L	7
H	L	L	H	L	L	L	H	H	H	H	H	H	H	8
H	L	L	H	L	L	H	H	H	H	L	H	H	H	9
H	L	L	H	L	H	L	L	L	L	L	L	L	L	blank
H	L	L	H	L	H	H	L	L	L	L	L	L	L	blank
H	L	L	H	H	L	L	L	L	L	L	L	L	L	blank
H	L	L	H	H	H	L	L	L	L	L	L	L	L	blank
H	L	L	H	H	H	H	L	L	L	L	L	L	L	blank
L	L	L	X	X	X	X	(5)							(5)
as above		H	as above				inverse of above							as above

Notes

1. H = HIGH state (the more positive voltage)
2. L = LOW state (the less positive voltage)
3. X = state is immaterial
4. For liquid crystal displays, apply a square-wave to PH.
For common cathode LED displays, select PH = LOW.
For common anode LED displays, select PH = HIGH.
5. Depends upon the BCD-code previously applied when LD = HIGH.



Program 8

Sterowanie multipleksowe

7 - segmentowych
wyświetlaczy LED ze wspólna
anodą wykorzystując dekodery
BCD 4543

```
D:\Mikroprocesory\Bascom Colege\basAVR_listingi\Dziala_zajecia\LED_45...
Sub
Label
$regfile = "m8def.dat"
$crystal = 8000000
Config Pind.0 = Output
Config Pind.1 = Output
Config Pind.2 = Output
Config Pind.3 = Output
Config Pind.7 = Output
Config Pinb.0 = Output
Config Pinb.1 = Output
Config Pinb.2 = Output
Config Pinb.3 = Output
Config Timer0 = Timer , Prescale = 256

Declare Sub Zapis_cyfry(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 Kr1 As Bit , Kr2 As Bit , Kr3 As Bit , Kr4 As Bit
Dim I As Byte

W1 Alias Portb.0
W2 Alias Portb.1
W3 Alias Portb.2
W4 Alias Portb.3
Kropki Alias Portd.7

Enable Interrupts
Enable Timer0
Load Timer0 , 125
Set Kr1
Set Kr2
Set Kr3
Set Kr4

Do
A = I
B = I
C = I
D = I
Incr I
If I = 11 Then
I = 0
End If
```

```
D:\Mikroprocesory\Bascom Colege\basAVR_listingi\8_6_bez_opis_zmien_C.bas
Label
$sim
$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 9
Call Wysw7seg(i)
Waitms 500
Next I
Loop
End

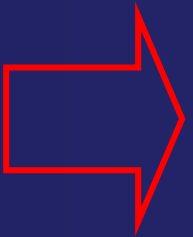
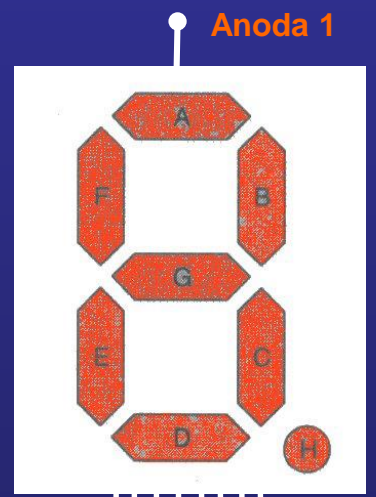
Sub Wysw7seg(byval C As Byte)
If C < 10 Then

Kod_cyf = Lookup(c , Kody7seg )

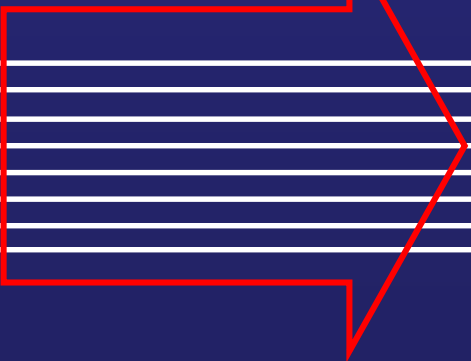
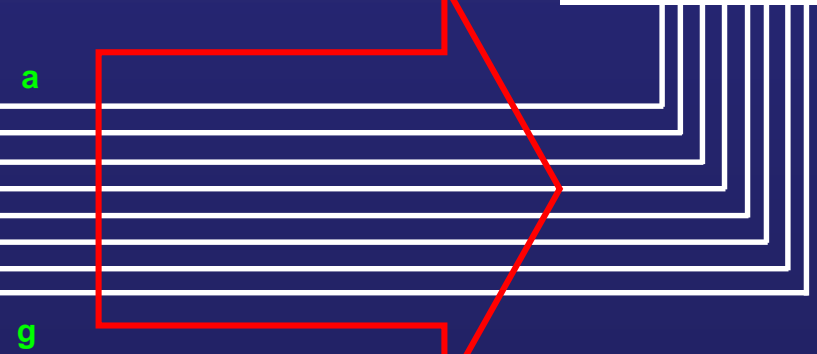
Kod_cyf.7 = Kropka
Portb = Kod_cyf
Else
Portd = Portb And &B10000000
End If
End Sub

Kody7seg:
Data &B00111111 , &B000000110 , &B01011011 , &B01001111 , &B01100110
Data &B01101101 , &B01111101 , &B000000111 , &B01111111 , &B01101111
```

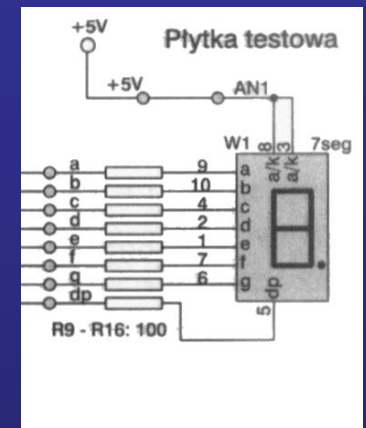
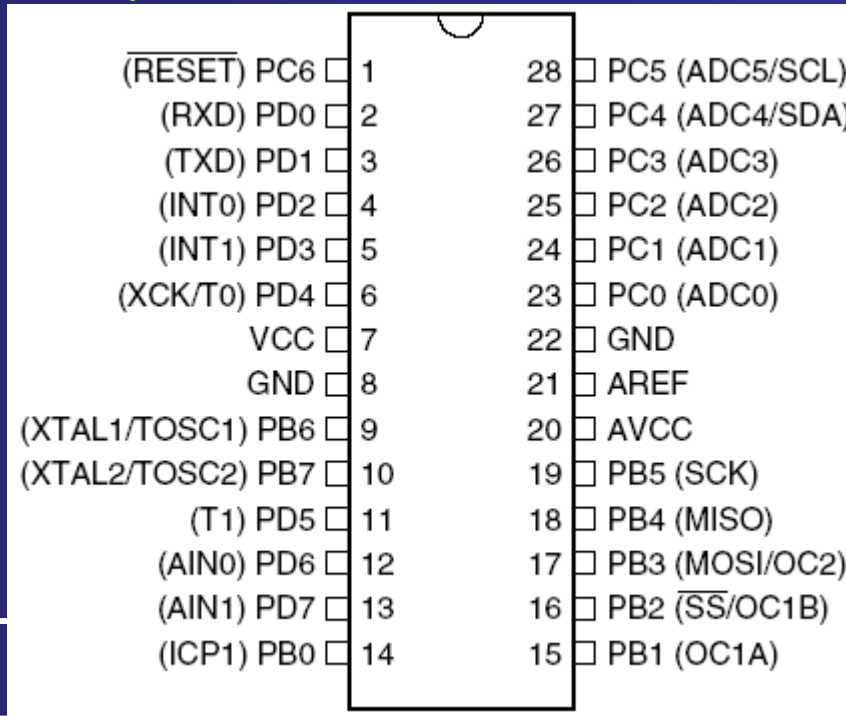
Dekoder kodu BCD na kod wyświetlacza 7 - segmentowego LED



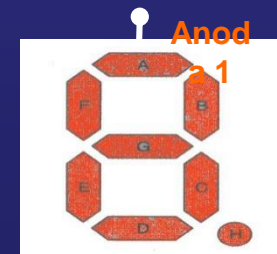
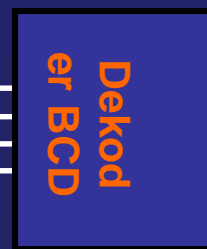
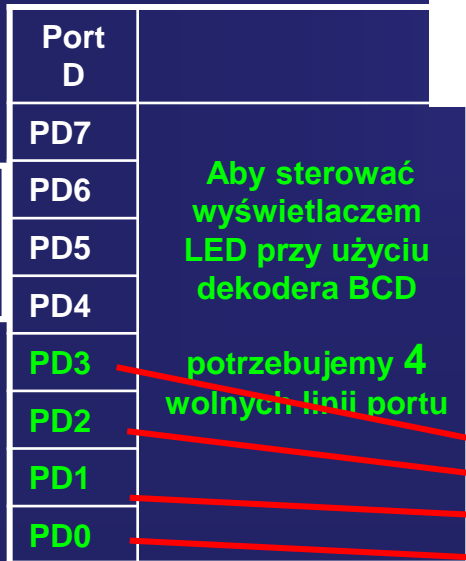
Dekoder BCD



Statyczne sterowanie 7 - segmentowego wyświetlacza LED



Inne urządzenie



potrzebujemy 4 wolnych linii portu

Aby sterować wyświetlaczem LED przy użyciu dekodera BCD

Program 8

Sterowanie multipleksowe

7 - segmentowych
wyświetlaczy LED ze wspólna
anodą wykorzystując dekodery
BCD 4543

```
D:\Mikroprocesory\Bascom Colege\basAVR_listingi\Dziala_zajecia\LED_45...
Sub
Toggle Kr1
Toggle Kr3
Wait 2

Loop
End

Sub Zapis_cyfry(cyfra As Byte)
Cyfra = Cyfra And &B00001111
Portd = Portd And &B11110000
Portd = Portd Or Cyfra
End Sub

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

Case 0:
Call Zapis_cyfry(a)
Kropki = Kr1
Reset W1
Case 1:
Call Zapis_cyfry(b)
Kropki = Kr2
Reset W2
Case 2:
Call Zapis_cyfry(c)
Kropki = Kr3
Reset W3
Case 3:
Call Zapis_cyfry(d)
Kropki = Kr4
Reset W4
End Select
Incr Nr_wysw

If Nr_wysw = 4 Then
Nr_wysw = 0
End If
Return
```

215

cyfra = 11010111

Cyfra = Cyfra And &B00001111

cyfra = 11010111

And $\frac{00001111}{00000111}$

cyfra = 11010111

or $\frac{00001111}{11011111}$

Wykorzystanie wyświetlaczy 7 - segmentowych LED

Zasilacz



Układ napyłniania cienkich warstw



Układ pomiaru próżni



Wykorzystanie wyświetlaczy 7 - segmentowych LED

