



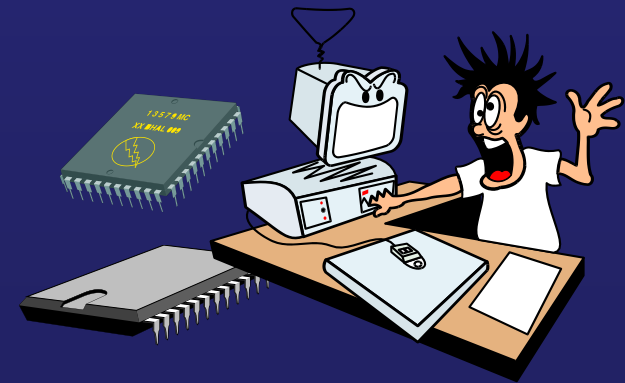
WYDZIAŁ FIZYKI
i INFORMATYKI STOSOWANEJ
Uniwersytet Łódzki



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

Systemy wbudowane

Kierunek: Informatyka
PRACOWNIA DYDAKTYCZNA

Wykład 6.

Obsługa przycisków i klawiatury

Przyciski i klawiatury

Ważnymi elementami towarzyszącymi mikrokontrolerom, oprócz wyświetlaczy są przyciski oraz klawiatury. Za pomocą tych elementów jest możliwe wpisanie do mikrokontrolera odpowiednich wartości lub przełączanie jego trybów pracy. Za ich pomocą użytkownik może wprowadzić dane lub modyfikować wartości konfigurowalnych parametrów programu.

Przycisk do mikrokontrolera można dołączyć bezpośrednio do jego wyprowadzeń lub dołączyć je matrycowo. Do obsługi klawiatury w układzie matrycowym potrzebna jest mniejsza liczba linii mikrokontrolera oraz odpowiedni program sekwencyjnie kontrolujący wciśnięcie poszczególnych przycisków.

Jeśli jest potrzebna bardzo rozbudowana klawiatura, to można dołączyć do mikrokontrolera komputerową klawiaturę AT, która ma ponad 101 klawiszy.

Przyciski i klawiatury

Do odczytywania stanu przycisku można zastosować procedurę z instrukcją warunkową **If...Then.**

Aby prawidłowo wykonać czynności po wciśnięciu przycisku należy:

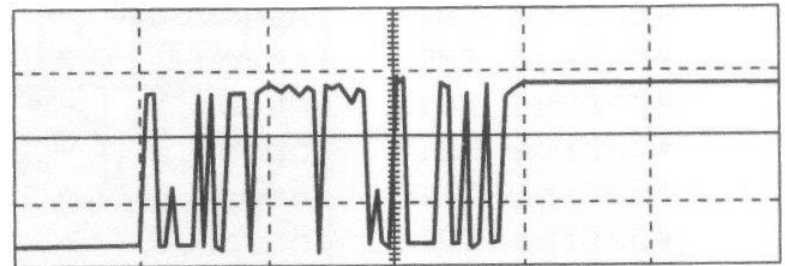
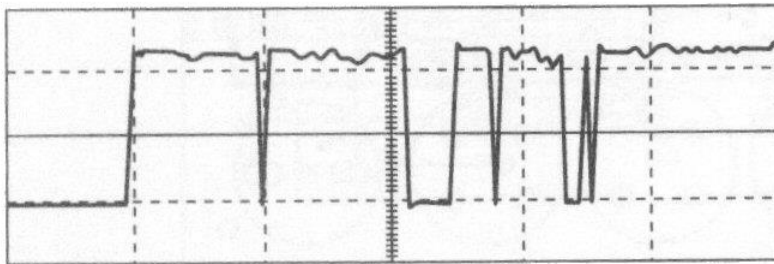
1. wykryć naciśnięcie przycisku;
2. odczekać 25 ms w celu eliminacji drgań styków;
3. ponownie sprawdzić, czy przycisk jest nadal wciśnięty; związane z tym funkcje;
4. czekać na puszczenie p
5. jeżeli wciśnięty, to wykonać rzycisku.

Przyciski i klawiatury

1 0 1 0 1 0 1 1 0 1 0 0 0

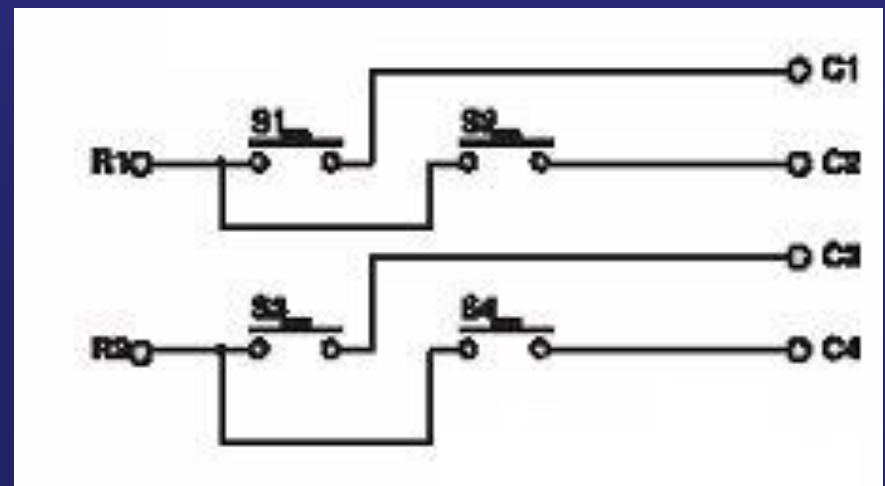
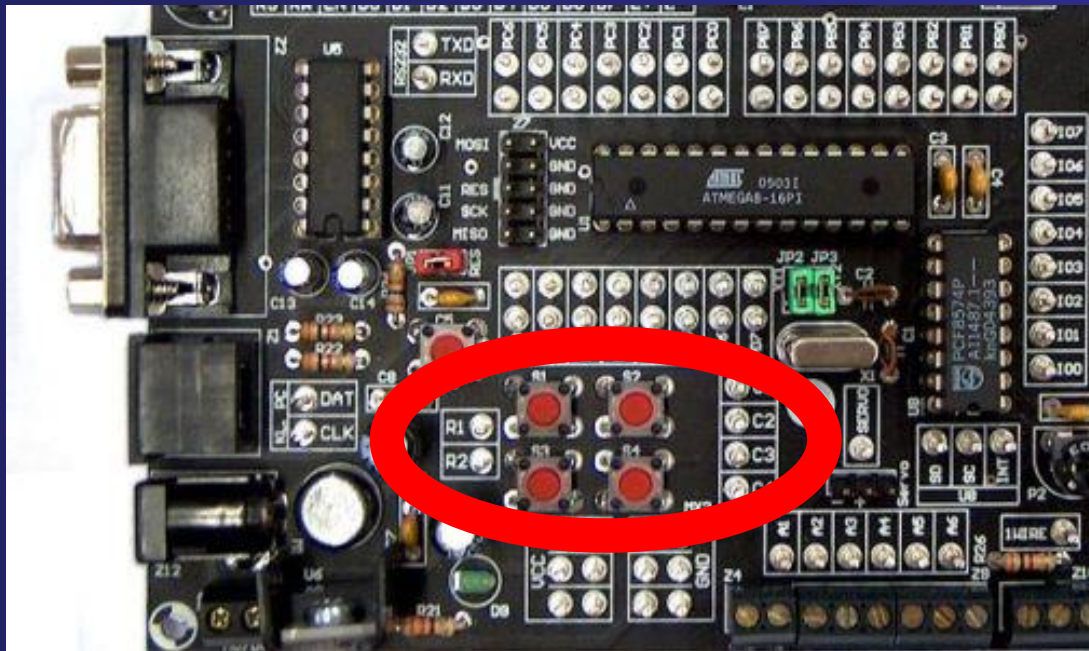
5 V

0 V



Przebiegi na linii wejściowej mikrokontrolera po naciśnięciu przycisku

Przyciski i klawiatury

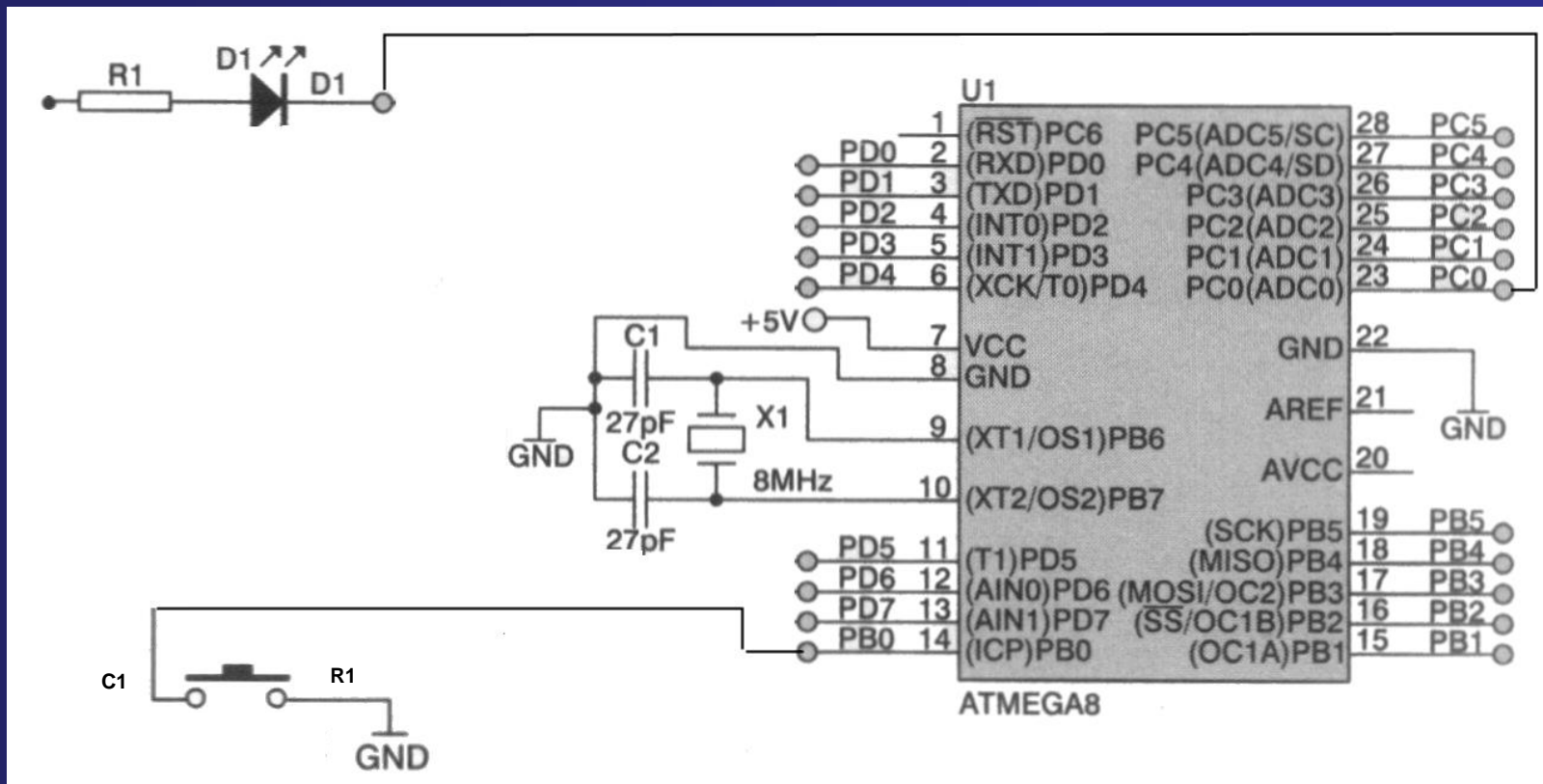


Program 11

Zastosowanie instrukcji warunkowej
If..Then
do odczytywania stanów przycisków

Program 11

Schemat połączenia diody LED do linii PBO portu B mikrokontrolera oraz przycisku S1 do linii PC0 portu C



Program 11

```
D:\Mikroprocesory\Bascom Colege\basA...
Sub
$regfile = "m8def.dat"
$crystal = 8000000

Config Pinb.0 = Input
Config Pinc.0 = Output

Set Portb.0

Do
  If Pinb.0 = 0 Then
    Reset Portc.0
  Else
    Set Portc.0
  End If
Loop
End
```

informuje kompilator o pliku dyrektyw mikrokontrolera

informuje kompilator o częstotliwości oscylatora taktującego mikrokontroler

linia PB0 jako wejściowa

linia PC0 jako wyjściowa

włączenie rezystora podciągającego na wejściu portu PB0

początek pętli

jeśli na wejściu portu PB0 występuje stan „0”, to ustaw wyjście portu PC0 na „0”

jeśli nie, czyli na wejściu portu PB0 występuje stan „1”, wówczas ustaw wyjście portu PC0 na „0”

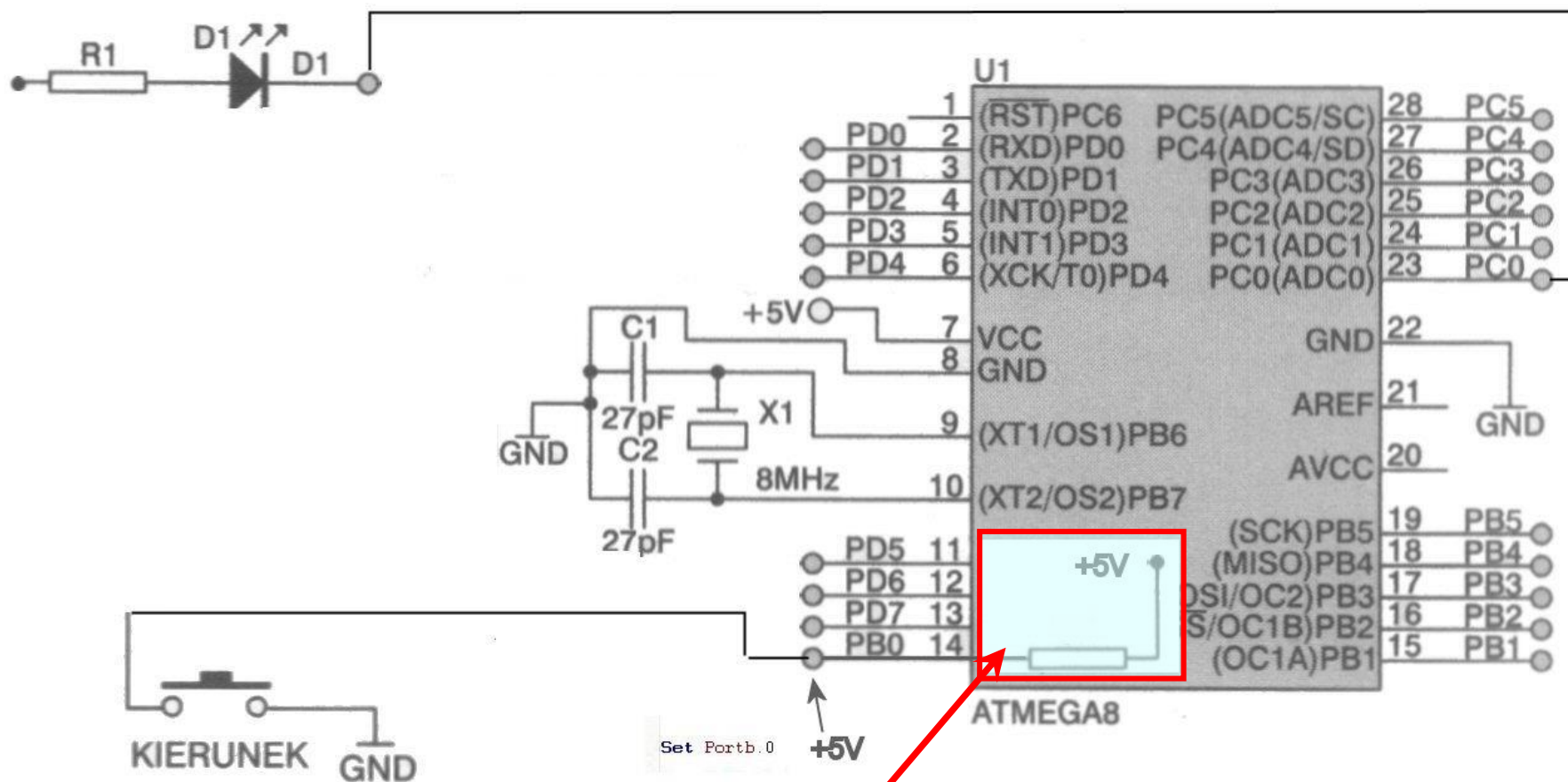
koniec warunku

koniec pętli głównej programu

koniec programu

Włączenie rezystora podciągającego na wejściu portu PB0

Set Portb.0



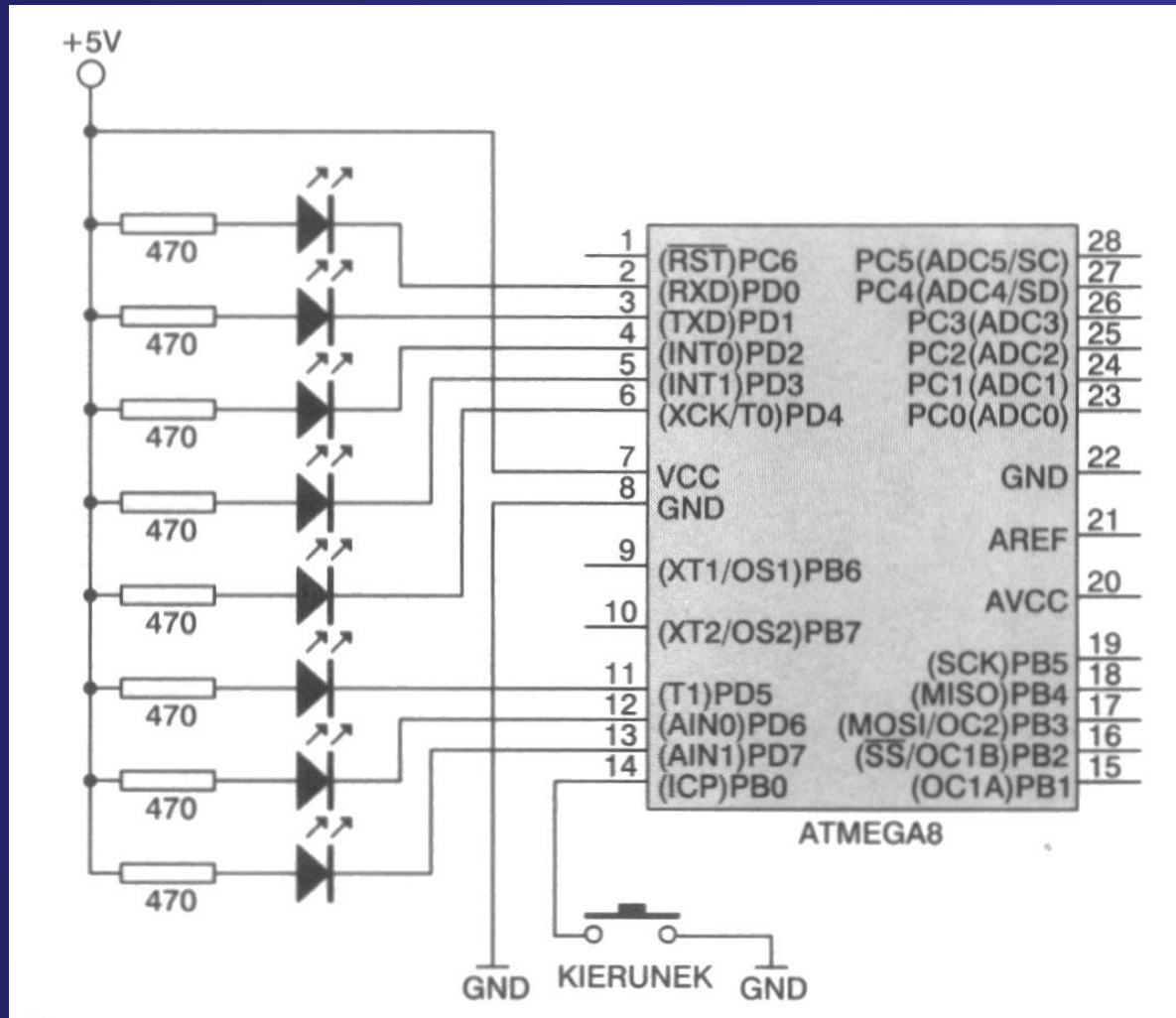
Set Portb.0

Program 12

Program sterujący diodami LED

Program 12

Schemat dołączenia diod LED oraz przycisku do mikrokontrolera



Program 12

```
D:\Mikroprocesory\Bascom Colege\basAVR_listingi\4...
Sub
Label

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

Config Portd = Output
Config Pinb.0 = Input

Dim Kierunek As Bit

Przycisk Alias Pinb.0

Set Portb.0

Portd = &B11111110
Reset Kierunek

Do
  If Przycisk = 0 Then
    Waitms 50
    Toggle Kierunek

    Do
      Loop Until Przycisk = 1

    End If

    If Kierunek = 1 Then
      Rotate Portd, Left
    Else
      Rotate Portd, Right
    End If

    Waitms 100
  Loop
End
```

wszystkie linie portu D jako wyjściowe

linia PB0 jako wyjściowa

deklaracja zmienne Kierunek jako bit

przypisanie zastępczej nazwy dla Pinb.0 - Przycisk

włączenie rezystora podciągającego na wejściu portu Pb0

wpisanie wartości binarnej do portu D, dzieś. 254

zerowanie zmiennej określającej kierunek zaświecania diod LED

jeśli przycisk naciśnięty, to odczekaj 50 ms i zmień stan na przeciwny zmiennej bitowej Kierunek

pętla wykonywana, aż do puszczenia przycisku

jeżeli zmienna Kierunek=„1”, to przesuwaj wpisane wartości do portu D w lewo

w przeciwnym wypadku przesuwaj wpisane wartości do portu D w prawo

czekaj 100 ms

koniec pętli głównej programu

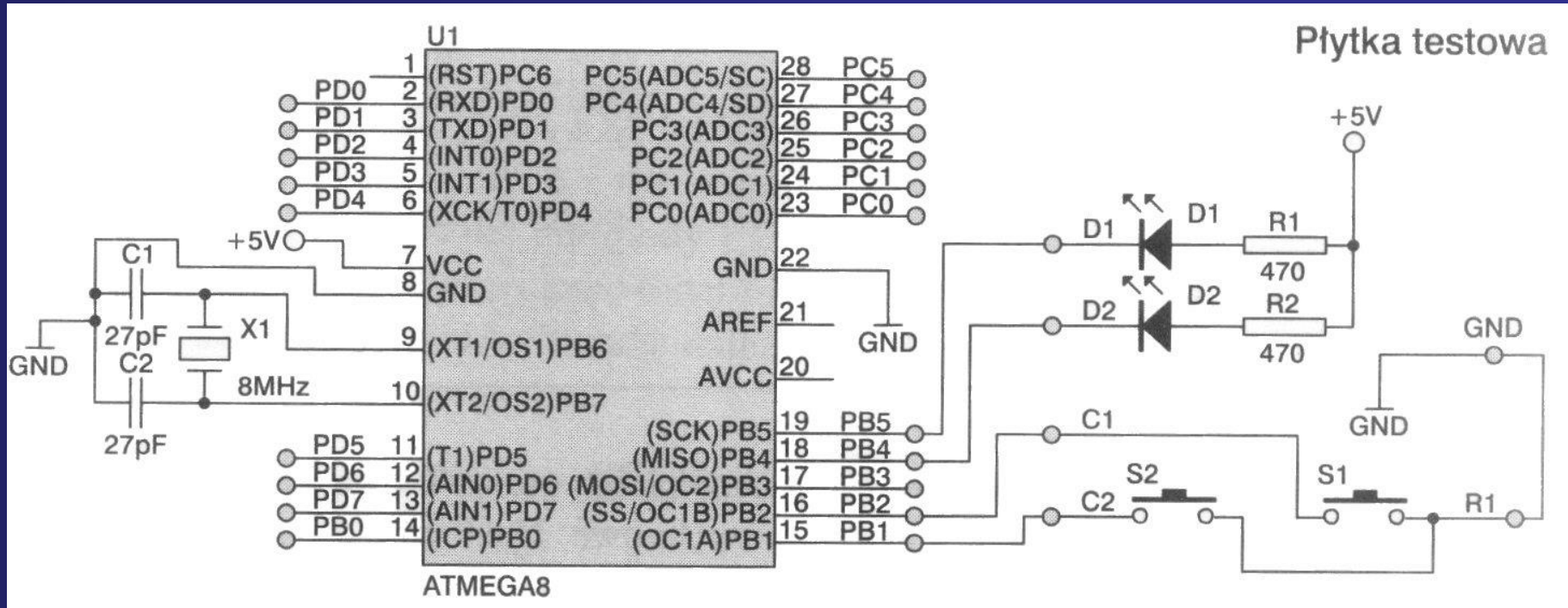
koniec programu

Program 13

Program obsługi przycisków S1, S2 za
pomocą instrukcji
Debounce

Program 13

Schemat dołączenia do mikrokontrolera przycisków oraz diod LED



Program 13

```
D:\Mikroprocesory\Bascom Colege\basAVR_listin...
Sub
Label

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

Config Pinb.1 = Input
Config Pinb.2 = Input
Config Pinb.4 = Output
Config Pinb.5 = Output

Led1 Alias Portb.5
Led2 Alias Portb.4
S1 Alias Pinb.2
S2 Alias Pinb.1

Set Portb.1
Set Portb.2

Do
  Debounce S1, 0, Pr1, Sub
  Debounce S2, 0, Pr2, Sub
Loop

End

Pr1:
  Toggle Led1
  Return

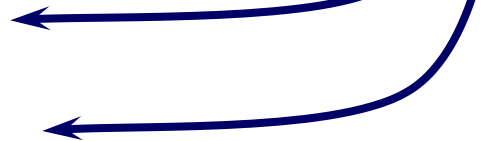
Pr2:
  Toggle Led2
  Return
```

Przypisanie nazw portom

włączenie rezystora podciągającego na wejściu
portu Pb1 i Pb2

jeśli naciśnięty przycisk S1, to skok do podprogramu Pr1

jeśli naciśnięty przycisk S2, to skok do podprogramu Pr2



Program 14

Program obsługi klawiatury
matrycowej 2x2 w przerwaniu zgłaszonym
od przepiętnienia Timer0

Klawiatura matrycowa

W sytuacji, gdy mamy niewiele wolnych linii mikrokontrolera, a potrzebujemy użyć wielu przycisków, można zastosować połączenie przycisków w układzie matrycowym. Przy zastosowaniu 9 przycisków, można je połączyć w matryce o 3 kolumnach i 3 wierszach. Takie połączenie wykorzystuje tylko 6 linii mikrokontrolera a nie 9, jak przy dołączeniu każdego przycisku do jednej linii portu.

W naszym przypadku klawiatura matrycowa 2x2 posiada 4 przyciski i wykorzystuje 4 linie mikrokontrolera. Oczywiście zastosowanie połączenia matrycowego zyskuje na znaczeniu gdy liczba przycisków jest większa niż 4.

Przykład:

3x3 daje 9 przycisków i wykorzystuje 6 linii mikrokontrolera

4x4 daje 16 przycisków i wykorzystuje 8 linii mikrokontrolera

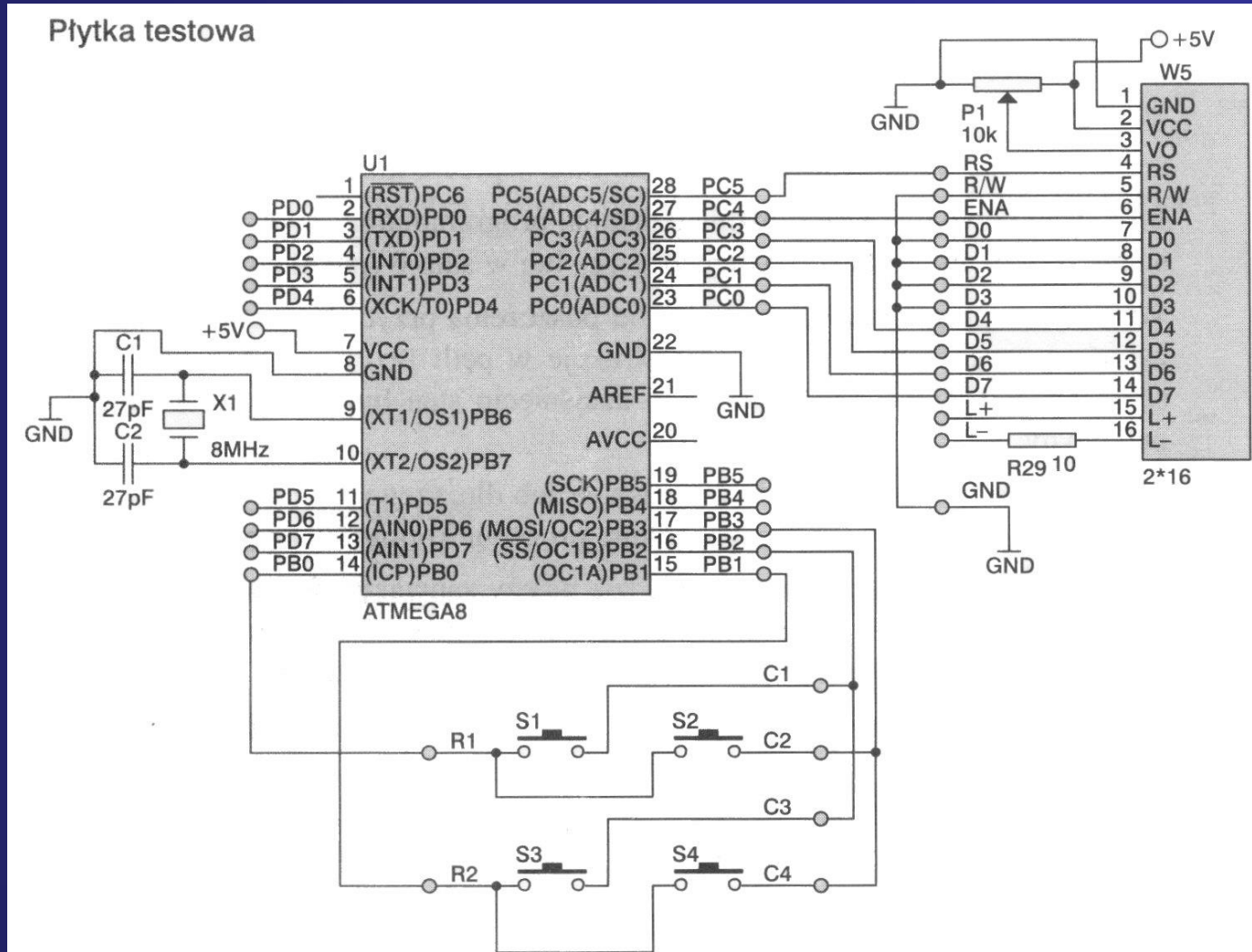
5x5 daje 25 przycisków i wykorzystuje 10 linii mikrokontrolera

Przy połączeniu matrycowym kolumny przycisków łączy się z liniami wyjściowymi, a wiersze są odczytywane przez linie wejściowe mikrokontrolera.

Obsługa matrycowej klawiatury jest zbliżona do sterowania multipleksowego wyświetlaczy LED. Odczyt klawiatury dobrze jest umieścić w podprogramie obsługi przerwania, aby był realizowany poza programem głównym.

Program 14

Schemat dołączenia klawiatury matrycowej do linii potu mikrokontrolera



Klawiatura matrycowa

1



Naciśnięcie
klawisza s1

Program 14

```
D:\Mikroprocesory\Bascom Colege\basAVR_listingi\Dziala_...
Sub                               Label
$regfile = "m8def.dat"
$crystal = 8000000

Config Pinb.0 = Input
Config Pinb.1 = Input
Config Pinb.2 = Output
Config Pinb.3 = Output
Config Lcd = 16 * 2
Config Timer0 = Timer , Prescale = 1024

On Timer0 Mult kl

Dim Temp1 As Byte
Dim Temp2 As Byte
Dim Przycisk As Byte
Dim I As Byte

W1 Alias Pinb.0
W2 Alias Pinb.1
Kol1 Alias Portb.2
Kol2 Alias Portb.3

Enable Interrupts
Enable Timer0
Load Timer0 , 200

Set Portb.0
Set Portb.1
Set Kol1
Set Kol2

Do
  Cls
  Lcd Przycisk
  Waitms 100
Loop
End
```

Konfiguracja portów, Timer0, wyświetlacza

Konfiguracja przerwania od przepełnienia Timer0

Przerwanie będzie generowane co 26ms

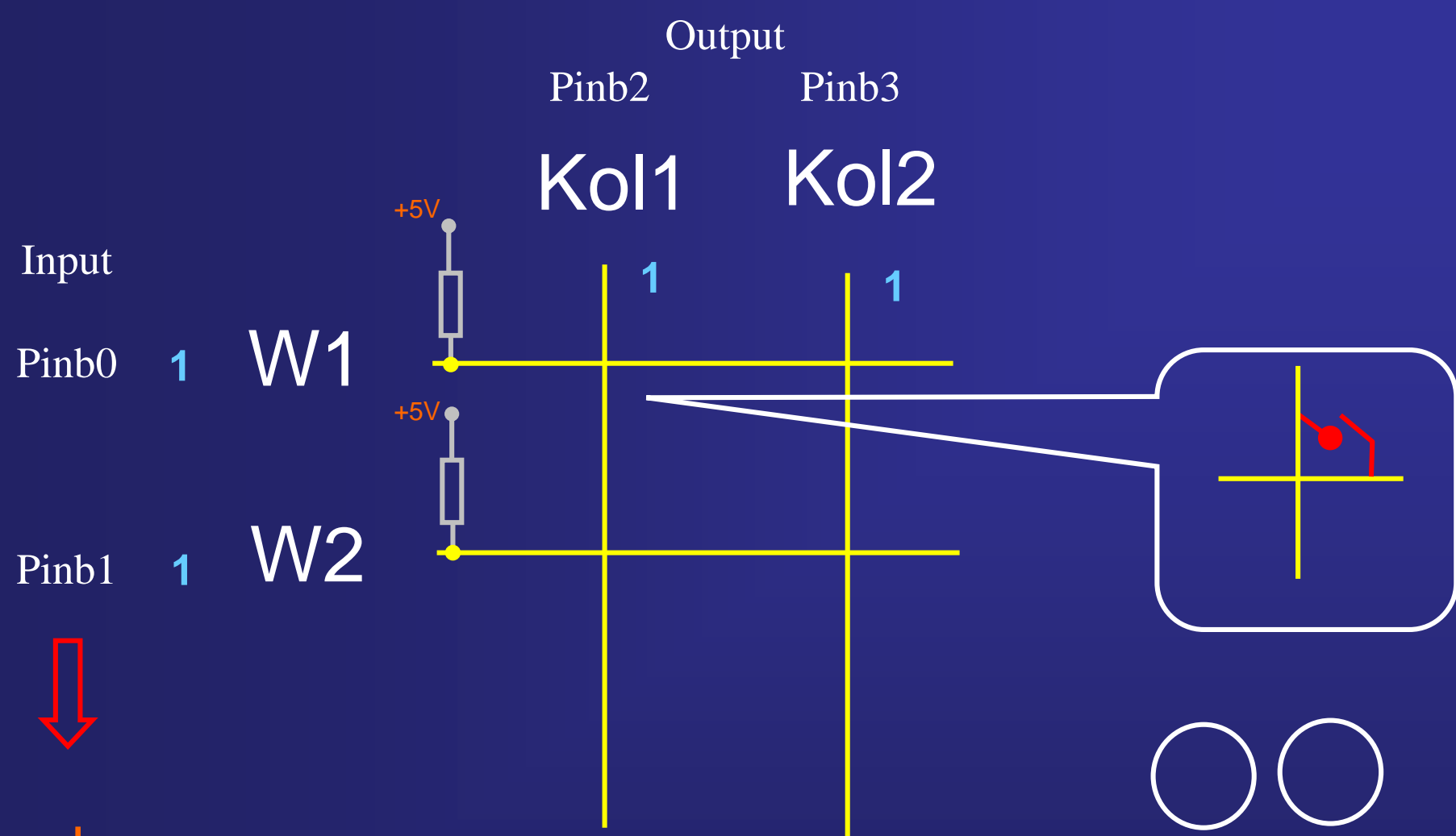
Deklaracja zmiennych

Przypisania nowych nazw portom

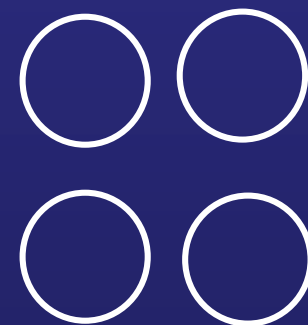
Dołączenie do portów rezystorów podciągających

Ustawienie „1” na portach Kol1, Kol2

Program główny wyświetla na LCD stan zmiennej Przycisk



**Klawiatura żaden
przycisk nie
wciśnięty**



Temp1 = bin 11 = 3

Output

Pinb2

Pinb3

Kol1

Kol2

+5V

+5V

Input

Pinb0

Pinb1

0

1

W1

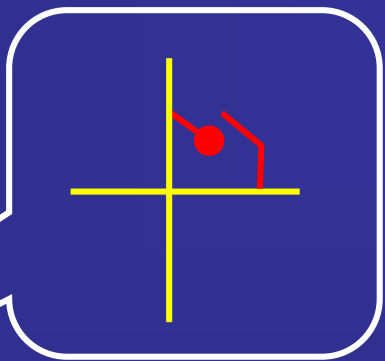
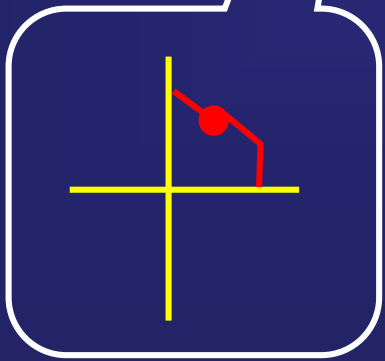
W2

0

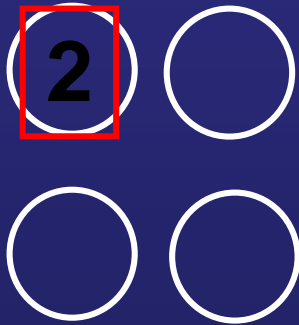
1



Temp1 = bin 10 = 2



Temp1 =
bin 10 = 2



Kol1 sprawdzana
w pierwszym
warunku pętli For

dla l=1

Kol2 sprawdzana
w drugim
warunku pętli For

dla l=2

Output

Pinb2

Pinb3

Kol1

Kol2

Input

Pinb0

Pinb1

1

0

W1

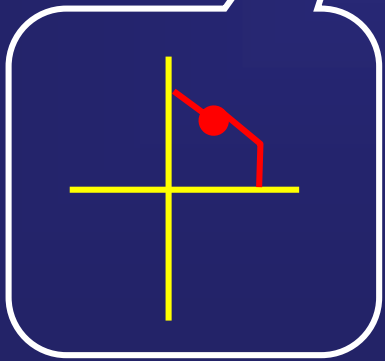
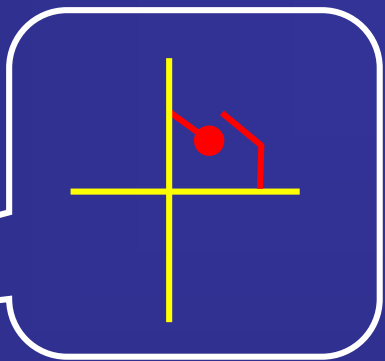
W2

+5V

+5V

0

1



Temp1 = bin 01 = 1

Temp1 =
bin 01 = 1

2

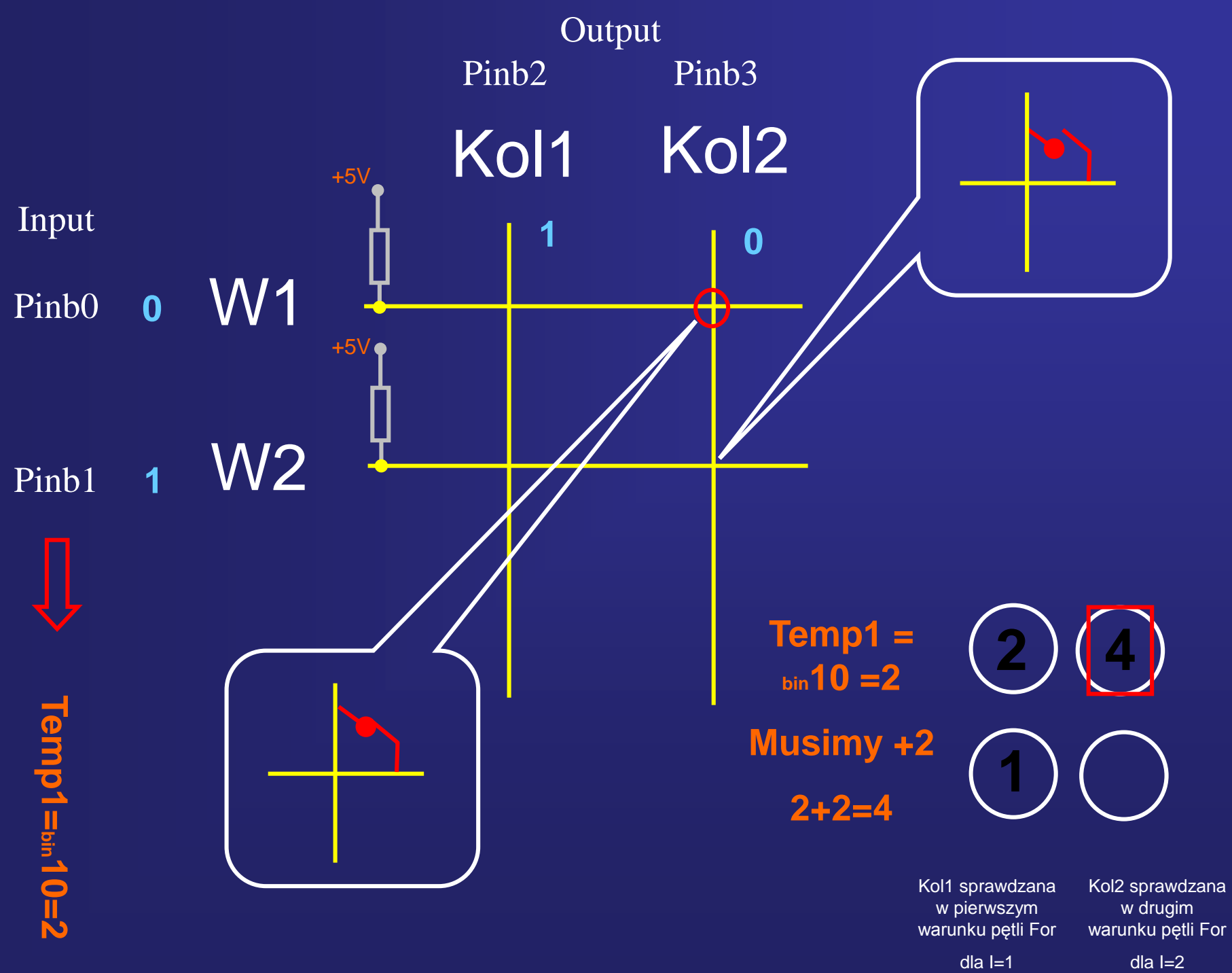
1

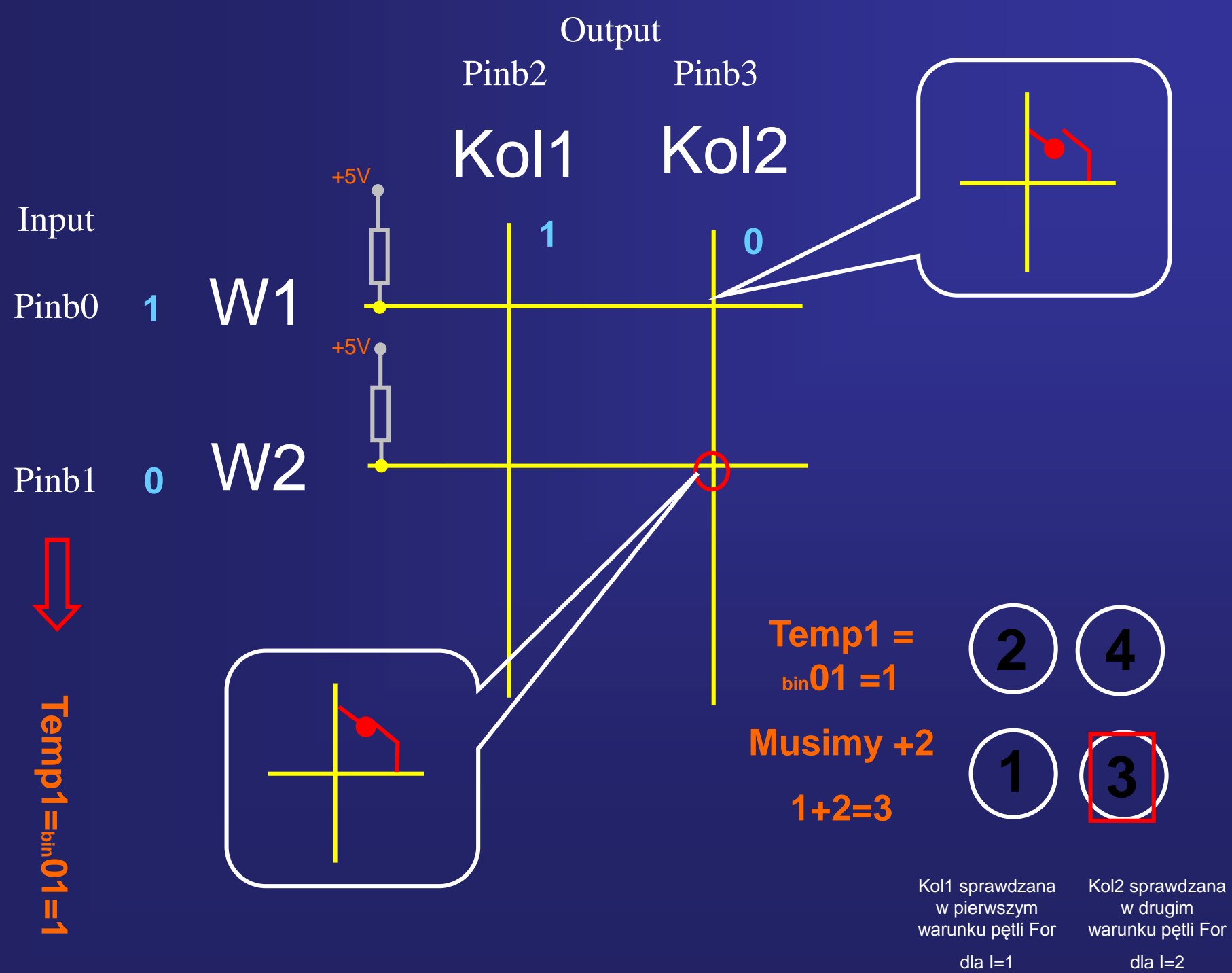
Kol1 sprawdzana
w pierwszym
warunku pętli For

Kol2 sprawdzana
w drugim
warunku pętli For

dla l=1

dla l=2





Program 14

```
Sub Sub Label  
Mult kl:  
Load Timer0 , 200  
For I = 1 To 2  
  If I = 1 Then  
    Reset Kol1  
  Else  
    Set Kol1  
    Reset Kol2  
  End If  
  If W1 = 0 Or W2 = 0 Then  
    Temp1 = Pinb And &B00000011  
  
    Exit For  
  Else  
    Temp1 = 0  
  End If  
Next I  
If Temp2 = Temp1 Then  
  Przycisk = Temp1  
  If I = 2 Then  
    Przycisk = Przycisk + 2  
  
  End If  
Else  
  Temp2 = Temp1  
  
End If  
Set Kol1  
Set Kol2  
Return
```

podprogram przerwania, w którym jest obsługiwana klawiatura

Pętla wykonywana 2 razy jeśli

I = 1, to zerowanie linii Kol1 w przeciwnym razie ustawienie „1” linii Kol1 i zerowanie linii Kol2

jeśli przy I = 1 lub I = 2 wejścia W1 lub W2 mają stany niskie - naciśnięty przycisk, to

zapisanie do zmiennej Temp1 wartości dwóch najmniej znaczących bitów rejestru PINB

- linii W1 i W2 i opuszczenie warunku

Jeśli war. jest niespełniony Temp1=0

Jeśli wartości w Temp2 oraz Temp1 są sobie równe, to zapisanie kodu naciśniętego przycisku do zmiennej Przycisk

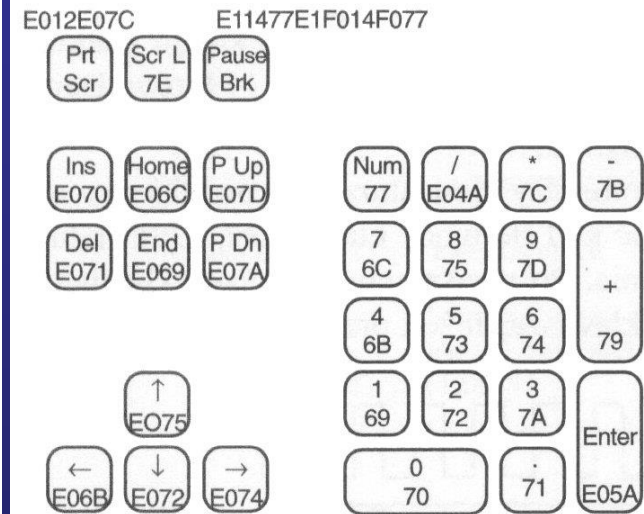
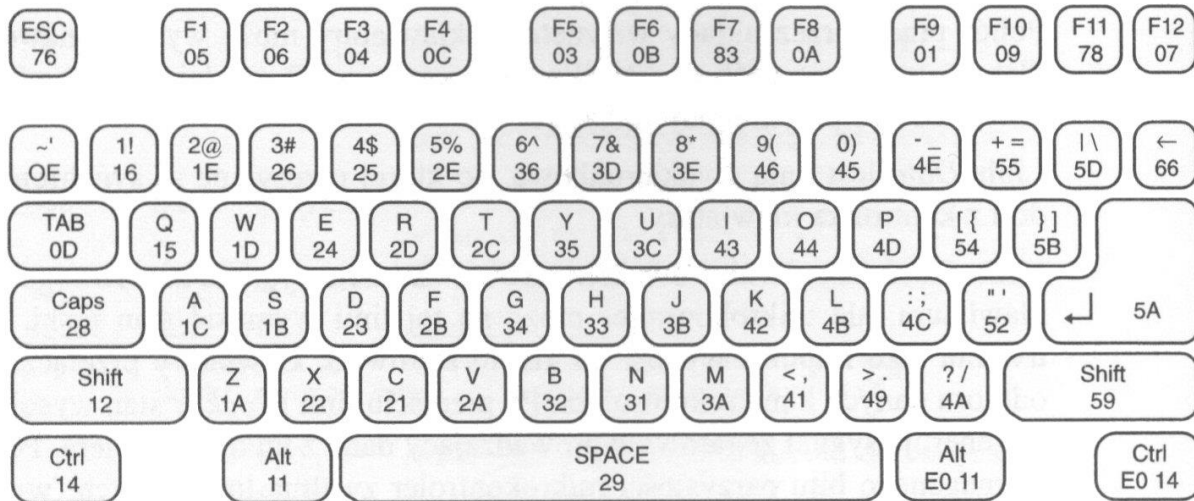
jeśli I = 2, to dodanie do zmiennej Przycisk wartości 2, gdyż odczytano stany przycisków drugiej wiersza

w przeciwnym razie zapisanie wartości Temp1 do zmiennej Temp2

Program 15

Obsługa komputerowej klawiatury AT

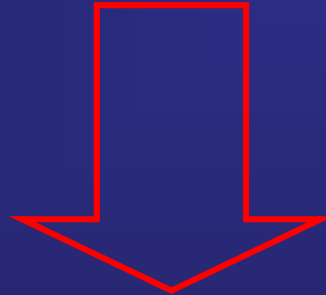
Kody podstawowych przycisków komputerowej klawiatury AT





Naciśnięcie klawisza generuje kod klawiatury

Na przykład: Literka „e” ⇒ kod 24



Konwertowanie kodu klawiatury na kod ASCII

Kod klawiatur 24 ⇒ kod ASCII 101

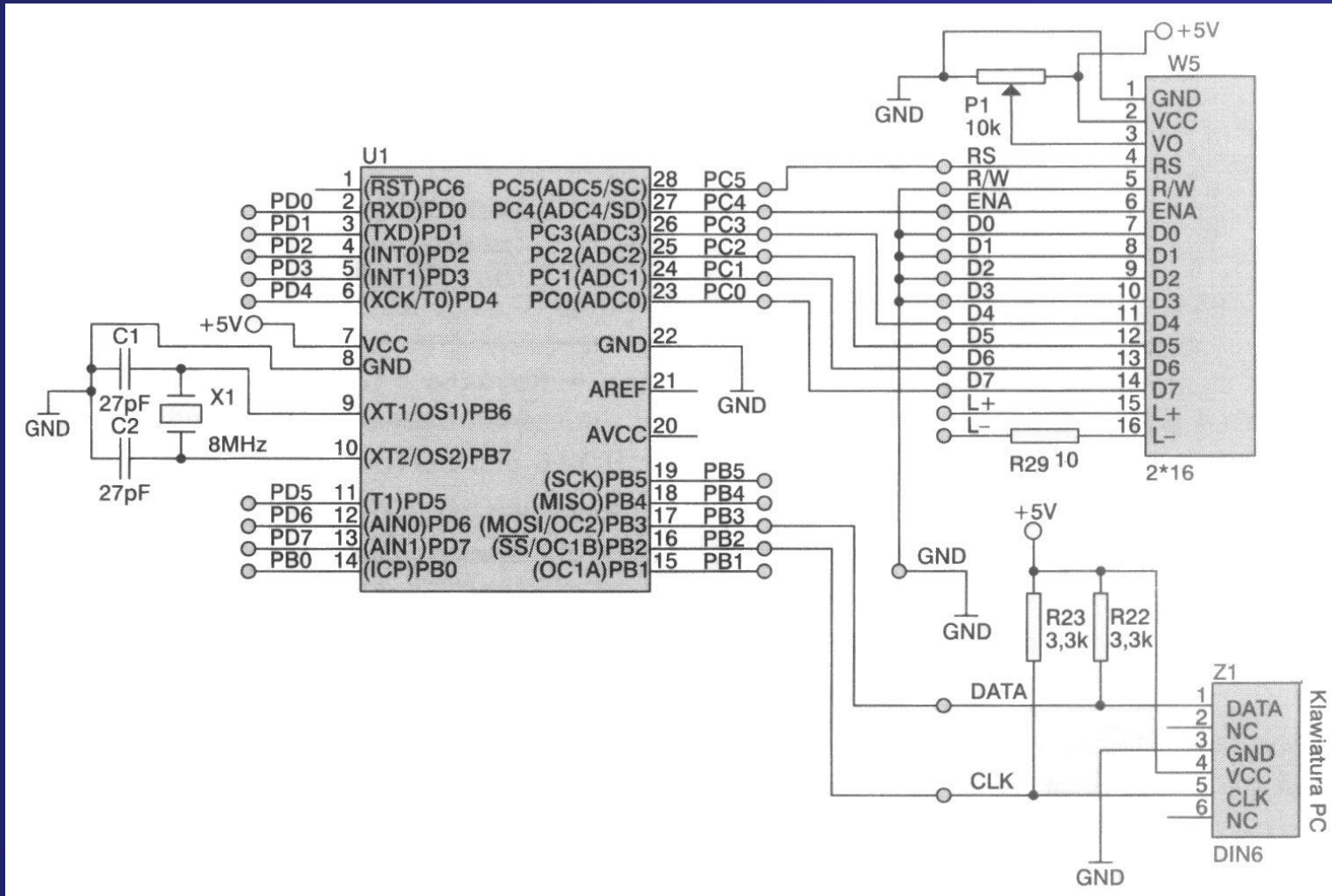
„e”

Kod ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Program 15

Schemat dołączenia komputerowej klawiatury AT do mikrokontrolera

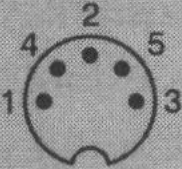



Dołączenia komputerowej klawiatury AT do zestawu *ZL2AVR*

złącze
PS/2 (DIN6)



Funkcje styków złączy AT (DIN5) oraz PS/2 (DIN6)

	 DIN41524, 5-pin DIN (AT)	 6-pin Mini DIN (PS2)
Zegar (CLK)	1	6
Linia danych (DATA)	2	1
Nie wykorzystywane (NC)	3	2, 5
GND	4	3
+5 V	5	4

Program 15

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

Config Keyboard = Pinb.2 , Data = Pinb.3 , Keydata = Keydata

Dim Kod_kl As Byte
Cls
Do
  Kod_kl = Getatkbd()
  If Kod_kl > 0 Then
    If Kod_kl > 31 And Kod_kl < 127 Then
      Lcd Chr(kod_kl)
    End If
    If Kod_kl = 44 Then
      Cls
    End If
    If Kod_kl = 13 Then
      Lowerline
    End If
  End If
Loop
End

Keydata:
  'klawisze normalnie - male litery
Data 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Data &H5E 0 0 0 0 0 0 0 113 49 0 0 0 122
Data 115 97 119 50 0 0 99 120 100 101 52
Data 51 0 0 32 118 102 116 114 53 0 0 110
Data 98 104 103 121 54 7 8 44 109 106 117 55
Data 56 0 0 44 107 105 111 48 57 0 0 46 45
Data 108 48 112 43 0 0 0 0 0 92 0 0 0
Data 0 13 0 0 92 0 0 0 60 0 0 0 8 0
Data 0 49 0 52 55 0 0 0 48 44 50 53 54 56
Data 0 0 0 43 51 45 42 57 0 0

  'klawisze z Shift - duze litery
Data 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Data 0 0 0 0 0 0 81 33 0 0 0 90 83 65
Data 87 34 0 0 67 88 68 69 0 35 0 0 32 86
Data 70 84 82 37 0 0 78 66 72 71 89 38 0
Data 0 76 77 74 85 47 40 0 0 59 75 73 79
Data 61 41 0 0 58 95 76 48 80 63 0 0 0 0
Data 0 0 96 0 0 0 0 13 94 0 42 0 0 0 62
Data 0 0 0 8 0 0 49 0 52 55 0 0 0 0 48
Data 44 50 53 54 56 0 0 0 43 51 45 42 57 0 0
```