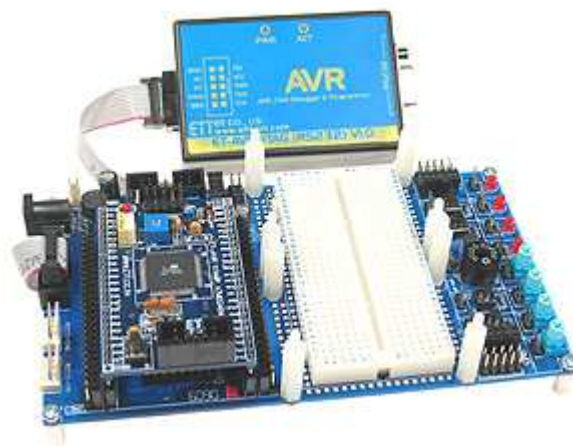


PRACOWNIA MIKROPROCESOROWA

Tomasz Fitzermann

ZESZYT ĆWICZEŃ



Poznań 2009

SPIS TREŚCI

1	NARZĘDZIA PROJEKTOWE	3
1.1	BASCOM AVR DEMO, BASCOM '51 DEMO	3
2	REJESTRY, ZMIENNE, PORTY WEJŚCIA-WYJŚCIA	6
2.1	SYSTEMY LICZBOWE, OPERACJA PRZYPISANIA	7
2.2	TYPY DANYCH, OPERACJE ARYTMETYCZNE I LOGICZNE	9
2.3	PROSTE UKŁADY CZASOWE, OPÓŹNIENIA, PĘTLE	11
2.4	PROGRAMOWA REALIZACJA FUNKCJI LOGICZNYCH	14
2.5	PRZERZUTNIKI, INSTRUKCJE WARUNKOWE	17
2.6	DEKODERY, KODERY, TRANSKODERY	19
2.7	DRIVERY - STEROWANIE SILNIKÓW KROKOWYCH	21
3	KOMUNIKACJA Z UŻYTKOWNIKIEM	24
3.1	WYŚWIETLACZ ALFANUMERYCZNY LCD	25
3.2	WYŚWIETLACZE LED	29
3.3	OBSŁUGA KŁAWIATUR	32
3.4	TRANSMISJA SZEREGOWA - UART, RS232	34
4	UKŁADY CZASOWE, PRZERWANIA	36
4.1	LICZNIK ZDARZEŃ	37
4.2	PRECYZYJNY GENERATOR	39
4.3	POMIAR CZĘSTOTLIWOŚCI	41
4.4	ZEGAR CYFROWY	43
4.5	PWM	46
4.6	RC5	49
5	KOMPARATOR ANALOGOWY	51
5.1	KOMPARATOR ANALOGOWY – PROSTY REGULATOR OŚWIETLENIA	52
6	PRZETWORNIKI A/C I C/A	54
6.1	PRZETWORNIK A/C – WOLTOMIERZ CYFROWY	55
7	I²C	57
7.1	PCF8591- KARTA PRZETWORNIKÓW A/C I C/A	58
7.2	AT24C04 – PAMIĘĆ EEPROM	60
8	1-WIRE	62
8.1	DS1820 - POMIAR TEMPERATURY	63
9	DODATKI	65
9.1	KODY ASCII	65

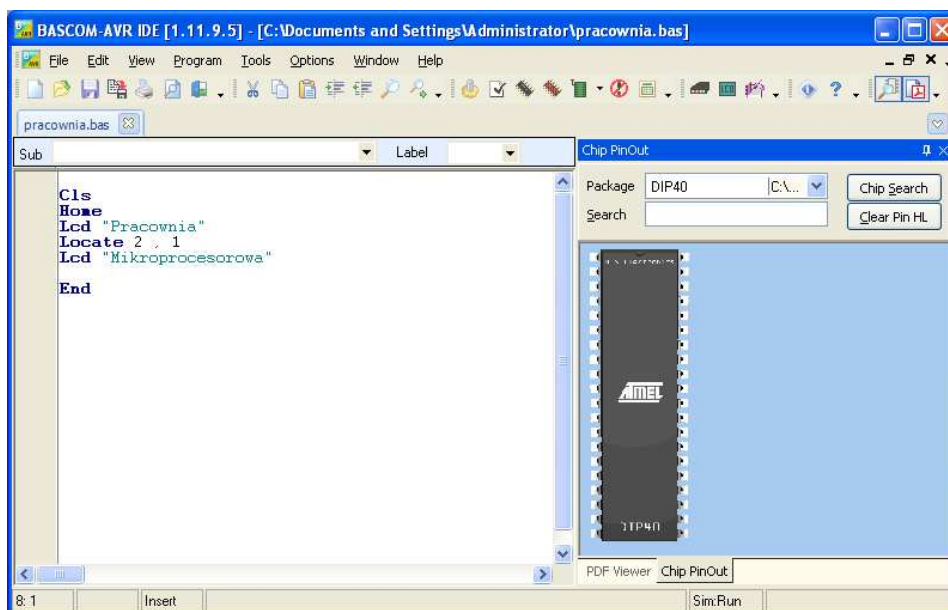
1 Narzędzia projektowe

1.1 BASCOM AVR DEMO, BASCOM '51 DEMO

Bascom AVR to zintegrowane środowisko programistyczne (IDE) przeznaczone do programowania mikrokontrolerów z rodziny AVR firmy ATMEL. Język ten jest zbliżony do języka Basic. Programowanie w języku Bascom jest stosunkowo łatwe, gdyż standardowo zawiera on bogatą bibliotekę funkcji do obsługi modułów peryferyjnych wbudowanych w mikrokontroler (m.in. przetwornik AC, łatwa konfiguracja i sterowanie linii portów we/wy, obsługa transmisji szeregowej, obsługa pamięci EEPROM itp.). Największą jednak zaletą Bascom AVR jest biblioteka funkcji zapewniających obsługę najczęściej wykorzystywanych układów w systemach mikroprocesorowych, np. znakowych i graficznych wyświetlaczy LCD, układów z interfejsami SPI, I2C, UART i 1-Wire (obsługa programowa i sprzętowa), klawiatury PC, klawiatury matrycowej oraz symulator umożliwiający zbadanie działania programu bez konieczności jego budowania.

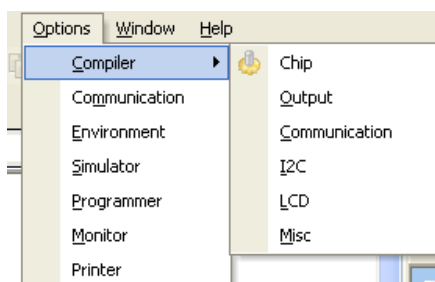
Środowisko BASCOM AVR składa się z kilku modułów:

 **Edytor** (kolorowanie składni, z możliwością podglądu wyprowadzeń mikrokontrolera)



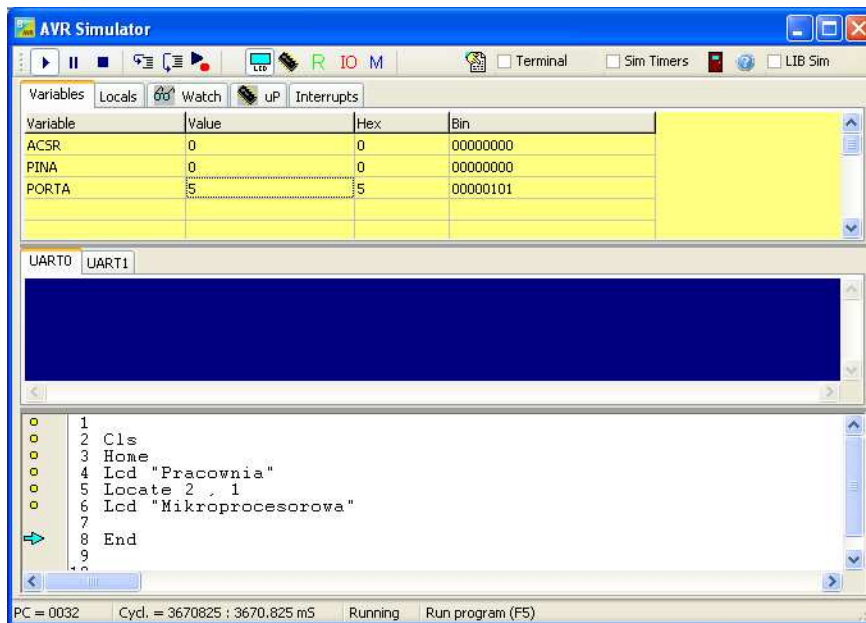
 **Kompilator**

W opcjach programu (Options) można w bardzo prosty sposób określić konfigurację kompilatora i urządzeń peryferyjnych

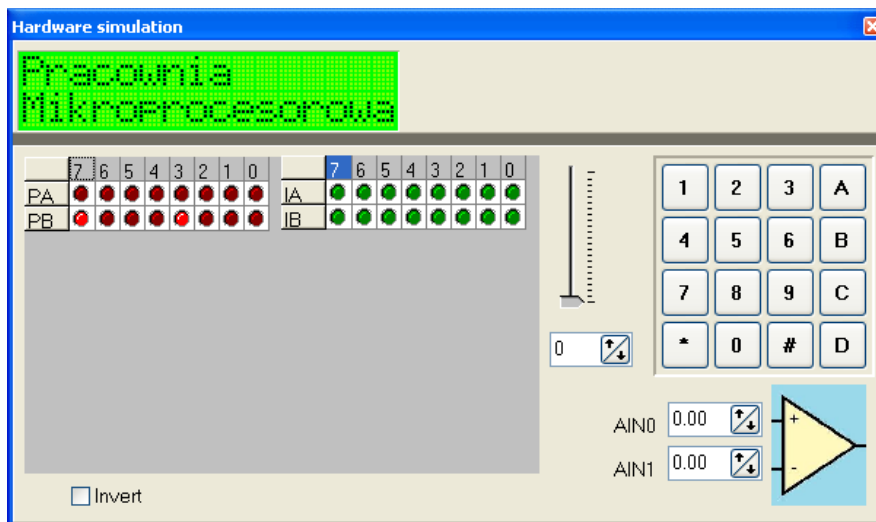




Symulator (praca krokowa, podgląd rejestrów i zmiennych)

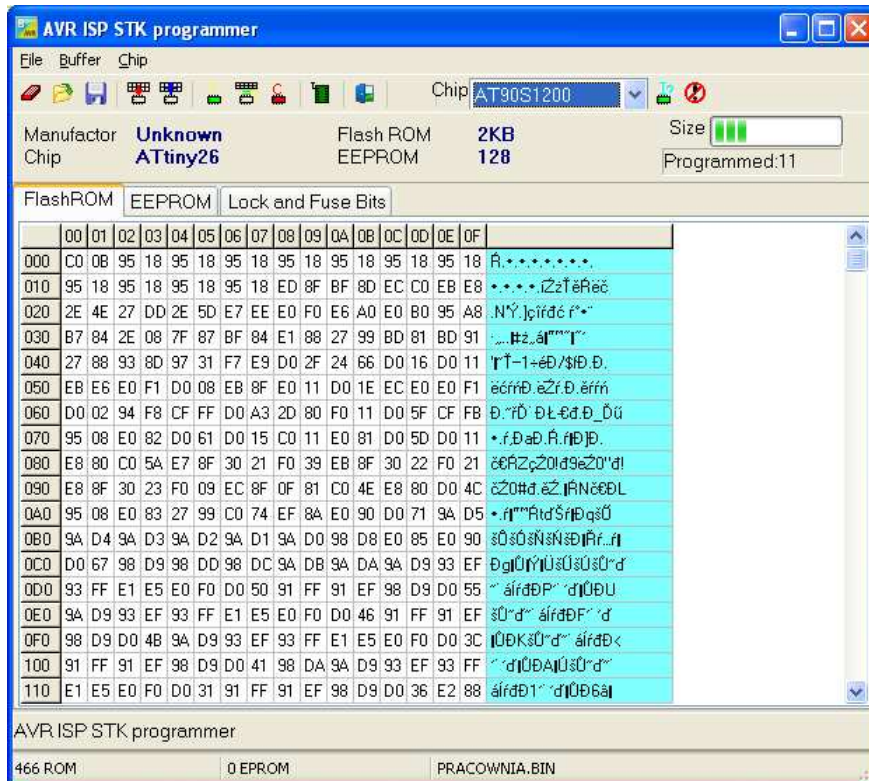


Symulator układów peryferyjnych (symulacja portów, komparatora, LCD, klawiatura, zadawanie napięć przetwornika A/C)





Programator



Do programowania mikrokontrolerów rodziny '51 wykorzystywane będzie środowisko BASCOM '51 DEMO o bardzo zbliżonych możliwościach. Podstawowym ograniczeniem wersji demonstracyjnych jest ograniczenie wielkości kompilowanego kodu do 2kB.

2 Rejestry, zmienne, porty wejścia-wyjścia

Porty mikrokontrolerów służą do komunikacji mikrokontrolera z zewnętrznymi układami peryferyjnymi (klawiatury, wskaźniki i wyświetlacze LED, LCD). Mikrokontrolery rodziny '51 posiadają od dwóch do czterech 8-bitowych uniwersalnych portów wejścia/wyjścia o oznaczeniach P0, P1, P2 i P3. Takie same nazwy mają rejestry umożliwiające dostęp do tych portów z poziomu programu. W przypadku mikrokontrolerów AVR dostępne 8-bitowe porty mają oznaczenia PA, PB, PC, PD. Rejestry dla portów wyjściowych to: PORTA, PORTB, PORTC i PORTD, natomiast rejestry wejściowe mają nazwy: PINA, PINB, PINC, PIND.

Tabela 2-1 Przykłady wyprowadzeń mikrokontrolerów w obudowach DIP-20 i DIP-40 z rodziny AVR i '51 stosowanych w zajęciach.

Rodzina '51	Rodzina AVR																																																																																																																																																																
AT89C4051 <table><tr><td>RST/VPP</td><td>1</td><td>20</td><td>VCC</td></tr><tr><td>(RXD) P3.0</td><td>2</td><td>19</td><td>P1.7</td></tr><tr><td>(TXD) P3.1</td><td>3</td><td>18</td><td>P1.6</td></tr><tr><td>XTAL2</td><td>4</td><td>17</td><td>P1.5</td></tr><tr><td>XTAL1</td><td>5</td><td>16</td><td>P1.4</td></tr><tr><td>(INT0) P3.2</td><td>6</td><td>15</td><td>P1.3</td></tr><tr><td>(INT1) P3.3</td><td>7</td><td>14</td><td>P1.2</td></tr><tr><td>(T0) P3.4</td><td>8</td><td>13</td><td>P1.1 (AIN1)</td></tr><tr><td>(T1) P3.5</td><td>9</td><td>12</td><td>P1.0 (AIN0)</td></tr><tr><td>GND</td><td>10</td><td>11</td><td>P3.7</td></tr></table>	RST/VPP	1	20	VCC	(RXD) P3.0	2	19	P1.7	(TXD) P3.1	3	18	P1.6	XTAL2	4	17	P1.5	XTAL1	5	16	P1.4	(INT0) P3.2	6	15	P1.3	(INT1) P3.3	7	14	P1.2	(T0) P3.4	8	13	P1.1 (AIN1)	(T1) P3.5	9	12	P1.0 (AIN0)	GND	10	11	P3.7	ATtiny26 <table><tr><td>(MOSI/DI/SDA/OC1A) PB0</td><td>1</td><td>20</td><td>PA0 (ADC0)</td></tr><tr><td>(MISO/DO/OC1A) PB1</td><td>2</td><td>19</td><td>PA1 (ADC1)</td></tr><tr><td>(SCK/SCL/OC1B) PB2</td><td>3</td><td>18</td><td>PA2 (ADC2)</td></tr><tr><td>(OC1B) PB3</td><td>4</td><td>17</td><td>PA3 (AREF)</td></tr><tr><td>VCC</td><td>5</td><td>16</td><td>GND</td></tr><tr><td>GND</td><td>6</td><td>15</td><td>AVCC</td></tr><tr><td>(ADC7/XTAL1) PB4</td><td>7</td><td>14</td><td>PA4 (ADC3)</td></tr><tr><td>(ADC8/XTAL2) PB5</td><td>8</td><td>13</td><td>PA5 (ADC4)</td></tr><tr><td>(ADC9/INT0/T0) PB6</td><td>9</td><td>12</td><td>PA6 (ADC5/AIN0)</td></tr><tr><td>(ADC10/RESET) PB7</td><td>10</td><td>11</td><td>PA7 (ADC6/AIN1)</td></tr></table>	(MOSI/DI/SDA/OC1A) PB0	1	20	PA0 (ADC0)	(MISO/DO/OC1A) PB1	2	19	PA1 (ADC1)	(SCK/SCL/OC1B) PB2	3	18	PA2 (ADC2)	(OC1B) PB3	4	17	PA3 (AREF)	VCC	5	16	GND	GND	6	15	AVCC	(ADC7/XTAL1) PB4	7	14	PA4 (ADC3)	(ADC8/XTAL2) PB5	8	13	PA5 (ADC4)	(ADC9/INT0/T0) PB6	9	12	PA6 (ADC5/AIN0)	(ADC10/RESET) PB7	10	11	PA7 (ADC6/AIN1)																																																																																
RST/VPP	1	20	VCC																																																																																																																																																														
(RXD) P3.0	2	19	P1.7																																																																																																																																																														
(TXD) P3.1	3	18	P1.6																																																																																																																																																														
XTAL2	4	17	P1.5																																																																																																																																																														
XTAL1	5	16	P1.4																																																																																																																																																														
(INT0) P3.2	6	15	P1.3																																																																																																																																																														
(INT1) P3.3	7	14	P1.2																																																																																																																																																														
(T0) P3.4	8	13	P1.1 (AIN1)																																																																																																																																																														
(T1) P3.5	9	12	P1.0 (AIN0)																																																																																																																																																														
GND	10	11	P3.7																																																																																																																																																														
(MOSI/DI/SDA/OC1A) PB0	1	20	PA0 (ADC0)																																																																																																																																																														
(MISO/DO/OC1A) PB1	2	19	PA1 (ADC1)																																																																																																																																																														
(SCK/SCL/OC1B) PB2	3	18	PA2 (ADC2)																																																																																																																																																														
(OC1B) PB3	4	17	PA3 (AREF)																																																																																																																																																														
VCC	5	16	GND																																																																																																																																																														
GND	6	15	AVCC																																																																																																																																																														
(ADC7/XTAL1) PB4	7	14	PA4 (ADC3)																																																																																																																																																														
(ADC8/XTAL2) PB5	8	13	PA5 (ADC4)																																																																																																																																																														
(ADC9/INT0/T0) PB6	9	12	PA6 (ADC5/AIN0)																																																																																																																																																														
(ADC10/RESET) PB7	10	11	PA7 (ADC6/AIN1)																																																																																																																																																														
AT89S8252 <table><tr><td>(T2) P1.0</td><td>1</td><td>40</td><td>VCC</td></tr><tr><td>(T2 EX) P1.1</td><td>2</td><td>39</td><td>P0.0 (AD0)</td></tr><tr><td>P1.2</td><td>3</td><td>38</td><td>P0.1 (AD1)</td></tr><tr><td>P1.3</td><td>4</td><td>37</td><td>P0.2 (AD2)</td></tr><tr><td>(SS) P1.4</td><td>5</td><td>36</td><td>P0.3 (AD3)</td></tr><tr><td>(MOSI) P1.5</td><td>6</td><td>35</td><td>P0.4 (AD4)</td></tr><tr><td>(MISO) P1.6</td><td>7</td><td>34</td><td>P0.5 (AD5)</td></tr><tr><td>(SCK) P1.7</td><td>8</td><td>33</td><td>P0.6 (AD6)</td></tr><tr><td>RST</td><td>9</td><td>32</td><td>P0.7 (AD7)</td></tr><tr><td>(RXD) P3.0</td><td>10</td><td>31</td><td>EA/VPP</td></tr><tr><td>(TXD) P3.1</td><td>11</td><td>30</td><td>ALE/PROG</td></tr><tr><td>(INT0) P3.2</td><td>12</td><td>29</td><td>PSEN</td></tr><tr><td>(INT1) P3.3</td><td>13</td><td>28</td><td>P2.7 (A15)</td></tr><tr><td>(T0) P3.4</td><td>14</td><td>27</td><td>P2.6 (A14)</td></tr><tr><td>(T1) P3.5</td><td>15</td><td>26</td><td>P2.5 (A13)</td></tr><tr><td>(WR) P3.6</td><td>16</td><td>25</td><td>P2.4 (A12)</td></tr><tr><td>(RD) P3.7</td><td>17</td><td>24</td><td>P2.3 (A11)</td></tr><tr><td>XTAL2</td><td>18</td><td>23</td><td>P2.2 (A10)</td></tr><tr><td>XTAL1</td><td>19</td><td>22</td><td>P2.1 (A9)</td></tr><tr><td>GND</td><td>20</td><td>21</td><td>P2.0 (A8)</td></tr></table>	(T2) P1.0	1	40	VCC	(T2 EX) P1.1	2	39	P0.0 (AD0)	P1.2	3	38	P0.1 (AD1)	P1.3	4	37	P0.2 (AD2)	(SS) P1.4	5	36	P0.3 (AD3)	(MOSI) P1.5	6	35	P0.4 (AD4)	(MISO) P1.6	7	34	P0.5 (AD5)	(SCK) P1.7	8	33	P0.6 (AD6)	RST	9	32	P0.7 (AD7)	(RXD) P3.0	10	31	EA/VPP	(TXD) P3.1	11	30	ALE/PROG	(INT0) P3.2	12	29	PSEN	(INT1) P3.3	13	28	P2.7 (A15)	(T0) P3.4	14	27	P2.6 (A14)	(T1) P3.5	15	26	P2.5 (A13)	(WR) P3.6	16	25	P2.4 (A12)	(RD) P3.7	17	24	P2.3 (A11)	XTAL2	18	23	P2.2 (A10)	XTAL1	19	22	P2.1 (A9)	GND	20	21	P2.0 (A8)	ATmega16 <table><tr><td>(XCK/T0) PB0</td><td>1</td><td>40</td><td>PA0 (ADC0)</td></tr><tr><td>(T1) PB1</td><td>2</td><td>39</td><td>PA1 (ADC1)</td></tr><tr><td>(INT2/AIN0) PB2</td><td>3</td><td>38</td><td>PA2 (ADC2)</td></tr><tr><td>(OC0/AIN1) PB3</td><td>4</td><td>37</td><td>PA3 (ADC3)</td></tr><tr><td>(SS) PB4</td><td>5</td><td>36</td><td>PA4 (ADC4)</td></tr><tr><td>(MOSI) PB5</td><td>6</td><td>35</td><td>PA5 (ADC5)</td></tr><tr><td>(MISO) PB6</td><td>7</td><td>34</td><td>PA6 (ADC6)</td></tr><tr><td>(SCK) PB7</td><td>8</td><td>33</td><td>PA7 (ADC7)</td></tr><tr><td>RESET</td><td>9</td><td>32</td><td>AREF</td></tr><tr><td>VCC</td><td>10</td><td>31</td><td>GND</td></tr><tr><td>GND</td><td>11</td><td>30</td><td>AVCC</td></tr><tr><td>XTAL2</td><td>12</td><td>29</td><td>PC7 (TOSC2)</td></tr><tr><td>XTAL1</td><td>13</td><td>28</td><td>PC6 (TOSC1)</td></tr><tr><td>(RXD) PD0</td><td>14</td><td>27</td><td>PC5 (TDI)</td></tr><tr><td>(TXD) PD1</td><td>15</td><td>26</td><td>PC4 (TDO)</td></tr><tr><td>(INT0) PD2</td><td>16</td><td>25</td><td>PC3 (TMS)</td></tr><tr><td>(INT1) PD3</td><td>17</td><td>24</td><td>PC2 (TCK)</td></tr><tr><td>(OC1B) PD4</td><td>18</td><td>23</td><td>PC1 (SDA)</td></tr><tr><td>(OC1A) PD5</td><td>19</td><td>22</td><td>PC0 (SCL)</td></tr><tr><td>(ICP1) PD6</td><td>20</td><td>21</td><td>PD7 (OC2)</td></tr></table>	(XCK/T0) PB0	1	40	PA0 (ADC0)	(T1) PB1	2	39	PA1 (ADC1)	(INT2/AIN0) PB2	3	38	PA2 (ADC2)	(OC0/AIN1) PB3	4	37	PA3 (ADC3)	(SS) PB4	5	36	PA4 (ADC4)	(MOSI) PB5	6	35	PA5 (ADC5)	(MISO) PB6	7	34	PA6 (ADC6)	(SCK) PB7	8	33	PA7 (ADC7)	RESET	9	32	AREF	VCC	10	31	GND	GND	11	30	AVCC	XTAL2	12	29	PC7 (TOSC2)	XTAL1	13	28	PC6 (TOSC1)	(RXD) PD0	14	27	PC5 (TDI)	(TXD) PD1	15	26	PC4 (TDO)	(INT0) PD2	16	25	PC3 (TMS)	(INT1) PD3	17	24	PC2 (TCK)	(OC1B) PD4	18	23	PC1 (SDA)	(OC1A) PD5	19	22	PC0 (SCL)	(ICP1) PD6	20	21	PD7 (OC2)
(T2) P1.0	1	40	VCC																																																																																																																																																														
(T2 EX) P1.1	2	39	P0.0 (AD0)																																																																																																																																																														
P1.2	3	38	P0.1 (AD1)																																																																																																																																																														
P1.3	4	37	P0.2 (AD2)																																																																																																																																																														
(SS) P1.4	5	36	P0.3 (AD3)																																																																																																																																																														
(MOSI) P1.5	6	35	P0.4 (AD4)																																																																																																																																																														
(MISO) P1.6	7	34	P0.5 (AD5)																																																																																																																																																														
(SCK) P1.7	8	33	P0.6 (AD6)																																																																																																																																																														
RST	9	32	P0.7 (AD7)																																																																																																																																																														
(RXD) P3.0	10	31	EA/VPP																																																																																																																																																														
(TXD) P3.1	11	30	ALE/PROG																																																																																																																																																														
(INT0) P3.2	12	29	PSEN																																																																																																																																																														
(INT1) P3.3	13	28	P2.7 (A15)																																																																																																																																																														
(T0) P3.4	14	27	P2.6 (A14)																																																																																																																																																														
(T1) P3.5	15	26	P2.5 (A13)																																																																																																																																																														
(WR) P3.6	16	25	P2.4 (A12)																																																																																																																																																														
(RD) P3.7	17	24	P2.3 (A11)																																																																																																																																																														
XTAL2	18	23	P2.2 (A10)																																																																																																																																																														
XTAL1	19	22	P2.1 (A9)																																																																																																																																																														
GND	20	21	P2.0 (A8)																																																																																																																																																														
(XCK/T0) PB0	1	40	PA0 (ADC0)																																																																																																																																																														
(T1) PB1	2	39	PA1 (ADC1)																																																																																																																																																														
(INT2/AIN0) PB2	3	38	PA2 (ADC2)																																																																																																																																																														
(OC0/AIN1) PB3	4	37	PA3 (ADC3)																																																																																																																																																														
(SS) PB4	5	36	PA4 (ADC4)																																																																																																																																																														
(MOSI) PB5	6	35	PA5 (ADC5)																																																																																																																																																														
(MISO) PB6	7	34	PA6 (ADC6)																																																																																																																																																														
(SCK) PB7	8	33	PA7 (ADC7)																																																																																																																																																														
RESET	9	32	AREF																																																																																																																																																														
VCC	10	31	GND																																																																																																																																																														
GND	11	30	AVCC																																																																																																																																																														
XTAL2	12	29	PC7 (TOSC2)																																																																																																																																																														
XTAL1	13	28	PC6 (TOSC1)																																																																																																																																																														
(RXD) PD0	14	27	PC5 (TDI)																																																																																																																																																														
(TXD) PD1	15	26	PC4 (TDO)																																																																																																																																																														
(INT0) PD2	16	25	PC3 (TMS)																																																																																																																																																														
(INT1) PD3	17	24	PC2 (TCK)																																																																																																																																																														
(OC1B) PD4	18	23	PC1 (SDA)																																																																																																																																																														
(OC1A) PD5	19	22	PC0 (SCL)																																																																																																																																																														
(ICP1) PD6	20	21	PD7 (OC2)																																																																																																																																																														



Większość linii portów może realizować kilka funkcji alternatywnych współpracujących z wbudowanymi układami np. wejścia przetworników A/C, linie przerwań np. INT0, INT1, wejścia liczników T0, T1 oraz linie układu transmisji szeregowej RxD i TxD.

2.1 Systemy liczbowe, operacja przypisania

Data:

Ocena:

i

Operacja przypisania powoduje umieszczenie wartości wyrażenia z prawej strony operatora = do zmiennej znajdującej się z lewej jego strony. W instrukcji przypisywania wartości zmiennej można używać zapisu dziesiętnego, heksadecymalnego bądź dwójkowego. Liczby zapisane szesnastkowo muszą być poprzedzone przedrostkiem &H, a zapis dwójkowy przedrostkiem &B. Poniższy przykład ukazuje tę samą liczbę w zapisie szesnastkowym, dwójkowym i dziesiętnym:

P0 = &HA
P1 = &B1010
P2 = 10



Uzupełnij tabelę.

BIN (2)	DEC (10)	HEX (16)
10100001		
	213	
		AC



Które bity portu P1 mają wartość „0” dla instrukcji P1 = 123.

P1 = &B_ _ _ _ _



Ustaw co drugą linię wyjściową portu P1 w stan „1”.

P1 = _ _ _

P1 = &B_ _ _ _ _

P1 = &H_ _

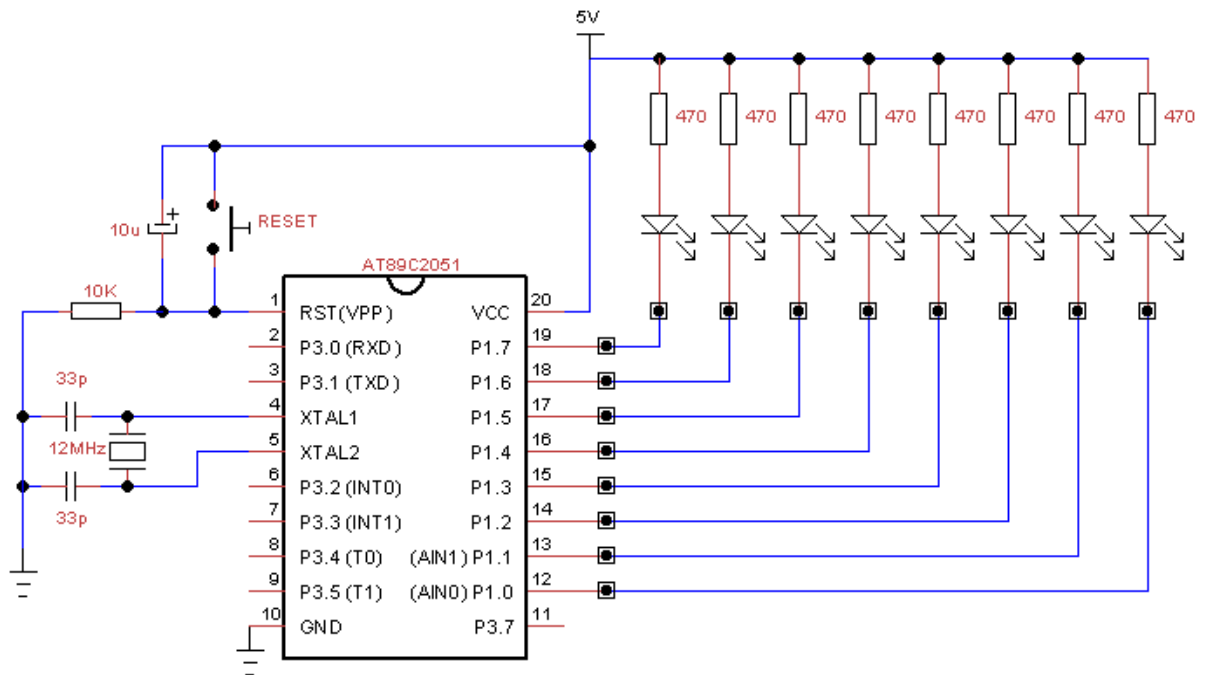


Skompiluj poniższy program . Następnie za pomocą symulatora  i symulatora układów peryferyjnych  sprawdź działanie instrukcji podstawienia danych do portów P0, P1, P2 i P3.

P0 = 10
P1 = 123
P2 = &B01010101
P3 = &HF0
End



Zaprogramuj układ 89C2051 i sprawdź działanie instrukcji w warunkach rzeczywistych dołączając diody LED do wyjść portu P1. Użyj wbudowanego modułu programatora - Programator MSC Electronics.



Program:

```
P1 = &B00001010
```

```
End
```



Dlaczego świecenie diod jest zanegowane w stosunku do wartości bitu?



2.2 Typy danych, operacje arytmetyczne i logiczne

Data:

Ocena:



W środowisku BASCOM można zdefiniować zmienne różnych typów. Typ określa rozmiar zajmowanej pamięci (bit, bajt, słowo 16bitowe itp.) oraz zakres wartości, które może przyjmować dana zmienna. Deklaracje zmiennych w programie wykonujemy za pomocą słowa kluczowego Dim:

Dim I As Integer, S As Single

Podstawowe typy danych języka BASCOM BASIC przedstawia poniższa tabela

Typ	Zajmowana pamięć	Uwagi
Bit	1 bit	Bit może przyjmować tylko dwie wartości: 0 i 1
Byte	1 bajt	Liczba całkowita z zakresu od 0 do 255
Integer	2 bajty	Liczba całkowita z zakresu -32768 do +32767
Long	4 bajty	Liczba całkowita z zakresu -2^{32} do $2^{32}-1$
Word	2 bajty	Liczba całkowita (nieujemna) z zakresu od 0 do 65535
Single	4 bajty	Liczba zmiennoprzecinkowa
String	max. 254 bajty	Typ String przechowuje dowolny ciąg znaków o długości nie większej niż 254 znaki. Każdy znak to jeden bajt. Ciąg zakończony jest zawsze znakiem 0, stąd tekst o długości 10 znaków zajmuje 11 bajtów.



Napisz poniższy program i sprawdź jego działanie za pomocą symulatora (praca krokowa, podgląd zmiennych). Uzupełnij odpowiednie pola wpisując wyniki operacji arytmetycznych. Sformułuj odpowiednie wnioski.

Dim X As Byte, Y As Integer, Z As Single

```

X = 10           ' X=
Y = X+10         ' Y=
Y = Y/3          ' Y=
Z = Y/3          ' Z=
Y = Y Mod 3      ' Y=
X = X*27         ' X=
  
```



.....

.....

.....

.....

.....

.....



Uzupełnij tabelę obliczając wynik wyrażeń związanych operatorami relacji.

Operator	Funkcja	X	Y	Wyrażenie	Wynik wyrażenia (prawda/fałsz)
=	Równość	1	2	$X = Y$	
<>	Nierówność	2	1	$X <> Y$	
<	Mniejszy	3	4	$X < Y$	
>	Większy	5	6	$X > Y$	
<=	Mniejszy lub równy	5	5	$X <= Y$	
>=	Większy lub równy	10	11	$X >= Y$	



Operatory logiczne stosuje się najczęściej jako łącznik w operacjach decyzyjnych (warunkowych), a także przy operacjach na bitach. Uzupełnij tabelę obliczając wynik operacji logicznych.

Operator	Znaczenie	X	Y	Z	Wyrażenie	Wynik wyrażenia (prawda/fałsz)
NOT	Logiczna negacja	2	3	4	NOT ($X < Y$)	
AND	Iloczyn logiczny	7	6	5	$(X < Y)$ AND ($Y < Z$)	
OR	Suma logiczna	10	5	10	$(X = Y)$ OR ($Y < Z$)	
XOR	Suma symetryczna	7	7	7	$(X = Y)$ XOR ($Y = Z$)	



Operacje na bitach. Operator AND jest często wykorzystywany przy maskowaniu bitów (zerowaniu poszczególnych bitów). Operator OR przy ustawianiu w stan 1 poszczególnych bitów, a operator XOR może zmieniać stan pojedynczych bitów na przeciwny. Uzupełnij tabelę.

Operator	X	$X_{(2)}$	Wyrażenie	Y=	$Y_{(2)}$
NOT	25		$Y = \text{NOT } X$		
AND	17		$Y = X \text{ AND } 15$		
OR	33		$Y = X \text{ OR } 15$		
XOR	47		$Y = X \text{ XOR } 15$		



.....

.....

.....

.....

.....

2.3 Proste układy czasowe, opóźnienia, pętle

Data:

Ocena:

Mikrokontrolery mogą wykonywać miliony operacji w ciągu sekundy i często wymagane jest opóźnienie wykonania kolejnych instrukcji dostosowując program do wolniejszych urządzeń zewnętrznych (np. wyświetlacz LCD). Uzależnienia czasowe można realizować za pomocą prostych funkcji opóźniających polegających na wielokrotnym powtarzaniu instrukcji o znanym czasie trwania. Jeśli potrzebna jest duża dokładność odmierzenia czasu opóźnienia, wtedy należy skorzystać z układów czasowo licznikowych pracujących jako Timery.

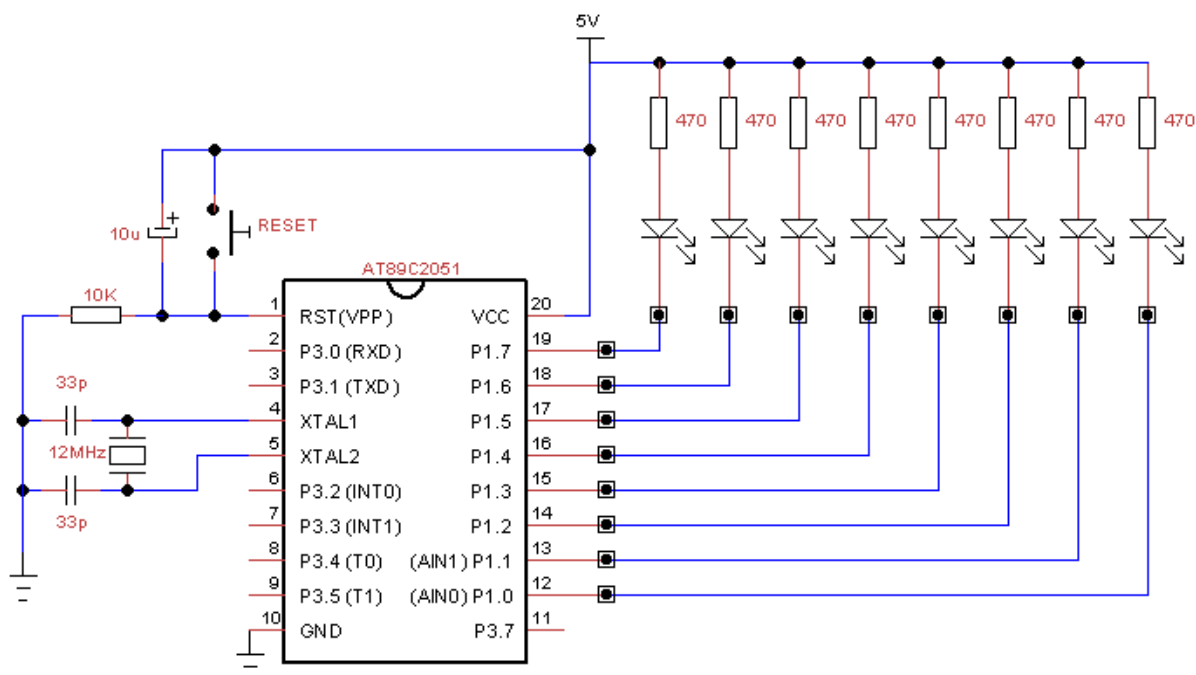


W środowisku BASCOM istnieje kilka przydatnych funkcji opóźniających

```
Delay      ' niewielkie opóźnienie (1[ms] dla AVR i 100[us] dla '51 przy
            'XTAL=12Mhz)
Waitms 100 ' opóźnienie ok. 100ms
Waitms A   ' opóźnienie ok A[ms], gdzie A to zmienna
Wait 2     ' opóźnienie ok 1[s]
```



Schemat układu pomiarowego





Przeanalizuj działanie programu realizującego „mruganie” diody LED podłączonej do linii P1.0.

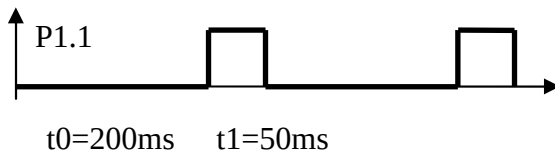
Do

```
P1.0 = 0  
waitms 100  
P1.0 = 1  
waitms 100
```

Loop



Napisz program wytwarzający przebieg na linii P1.1 podany na poniższym rysunku.



Oblicz częstotliwość przebiegu uzyskanego w poprzednim przykładzie

$f = \dots\dots\dots$



Zmieniaj okres i współczynnik wypełnienia fali obserwując wpływ tych parametrów na efekt świecenia diody LED. Sformułuj odpowiednie wnioski.

.....

.....

.....



Zapoznaj się z instrukcją SHIFT. Napisz program realizujący przesuwanie w prawo lub lewo kombinacji diod tzw. linijkę świetlną.



.....

.....

.....

.....

.....

2.4 Programowa realizacja funkcji logicznych

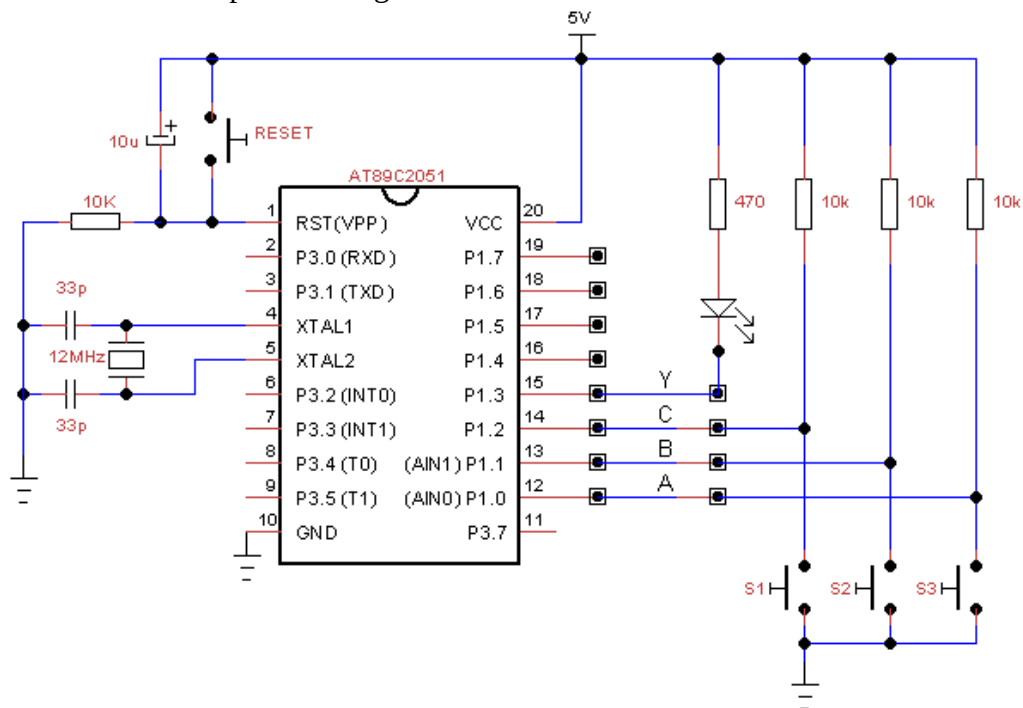
Data:

Ocena:

Programowa realizacja funkcji logicznych polega na cyklicznym wykonywaniu ciągu operacji bitowych lub warunkowych na poszczególnych liniach wejściowych i generowaniu wyniku na jednej bądź wielu liniach wyjściowych.



Schemat układu pomiarowego



Przykład programu realizującego funkcję logiczną $Y = \sim(AB+C)$. Wejściom przyporządkowano linie: A=P1.0, B=P1.1 i C=P1.2, wyjście Y realizuje linia P1.3.

```
Dim Y As bit
```

```
Do
```

```
    Y = P1.0 AND P1.1
```

```
    Y = Y OR P1.2
```

```
    P1.3 = NOT Y
```

```
Loop
```



Narysuj za pomocą symboli bramek logicznych układ realizujący funkcję z poprzedniego ćwiczenia.



Narysuj za pomocą symboli bramek logicznych układ realizujący funkcję zadaną przez prowadzącego. Uruchom i przeanalizuj działanie programu realizującego tę funkcję.

Funkcja:

$Y =$

Schemat układu:

Program:



Uruchom i przeanalizuj działanie programu realizującego funkcję logiczną polegającą na ustawieniu wyjścia w stan „0” jeśli na dokładnie jednym wejściu będzie stan „1”.

Wejścia			Wyjście
P1.2	P1.1	P1.0	P1.3
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

```

Dim Y As bit
Do
    Y = 1
    If P1.2=0 AND P1.1=0 AND P1.0=1 Then Y=0
    If P1.2=0 AND P1.1=1 AND P1.0=0 Then Y=0
    If P1.2=1 AND P1.1=0 AND P1.0=0 Then Y=0
    P1.3 = Y
Loop

```



Napisz program realizujący funkcję logiczną opisaną tabelą prawdy zadaną przez prowadzącego.

Wejścia			Wyjście
P1.2	P1.1	P1.0	P1.3
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



.....

.....

.....

.....

.....

2.5 Przerzutniki, instrukcje warunkowe

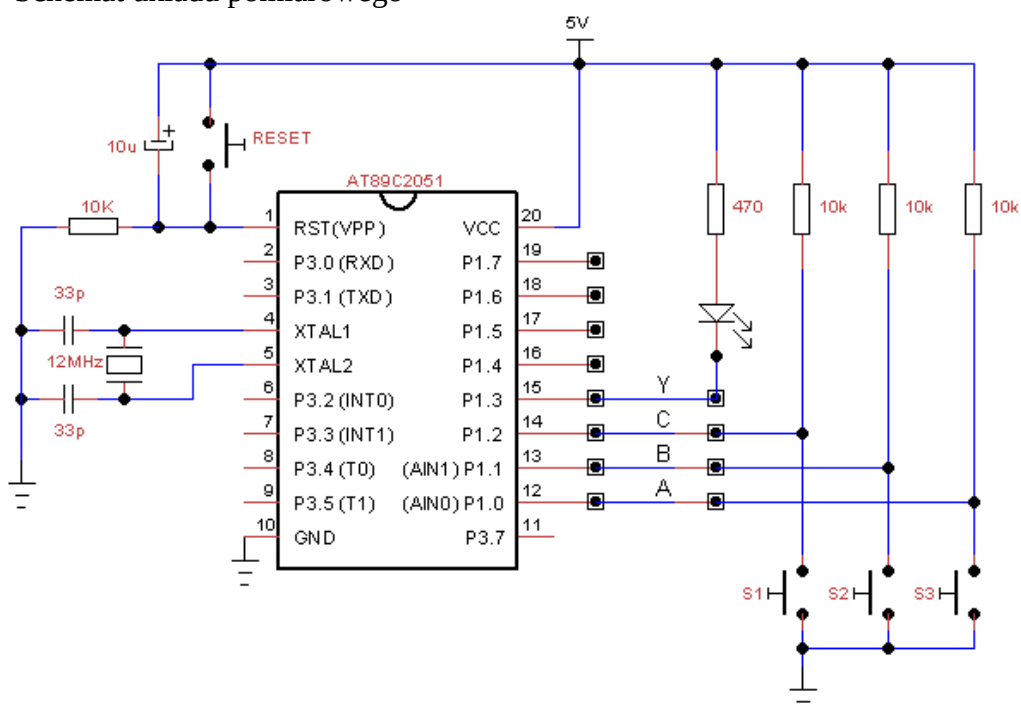
Data:

Ocena:

Programowa realizacja przerzutników polega na cyklicznym wykonywaniu ciągu operacji bitowych lub warunkowych na poszczególnych liniach wejściowych, zapamiętywaniu stanów i generowaniu wyniku na jednej bądź wielu liniach wyjściowych.



Schemat układu pomiarowego

*i*

Realizacja przerzutnika monostabilnego. Wejście wyzwalające - P1.0. Czas stanu niestabilnego 5 sekund.

Do

```
P1.3 = 1      ' wyłączenie diody LED
If P1.0 = 0 Then
    P1.3 = 0  ' wyzwolenie
    Wait 5    ' włączenie diody LED
Endif
```

Loop



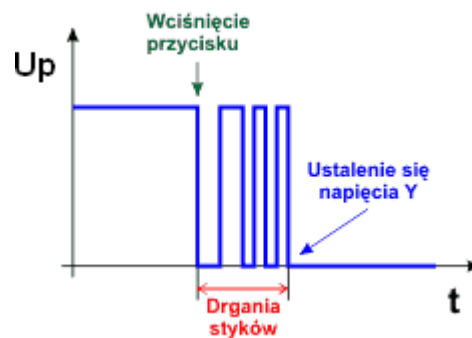
Podaj przykłady zastosowań przerzutników monostabilnych



Napisz program realizujący przerzutnik bistabilny (dwustabilny - podanie impulsu na wejście wyzwala zmianę stanu wyjścia na przeciwny i pozostawanie w tym stanie, aż do następnego wyzwolenia).



Zmodyfikuj poprzedni program eliminując wpływ drgań zestyków (podpowiedź – zastosuj niewielkie opóźnienie)



.....

.....

.....

.....

.....

2.6 Dekodery, kodery, transkodery

Data:

Ocena:

W przesyłaniu sygnałów cyfrowych istnieje zwykle potrzeba przetwarzania informacji z jednej zakodowanej postaci w drugą. Do tego celu służą specjalne układy cyfrowe zwane konwerterami kodów. Układy służące do przetwarzania kodu "1 z n" na dowolny inny kod dwójkowy przyjęto nazywać koderami, natomiast układy wykonujące operację odwrotną - dekodernami. Układy przetwarzające dowolny kod dwójkowy na inny, lecz nie "1 z n" nazywa się ogólnie transkoderami. Układy logiczne koderów, dekodów i konwerterów są układami kombinacyjnymi.

i

Realizacja układów kombinacyjnych za pomocą tablicy przeglądowej lookup.

Wejścia		Wyjścia			
P3.1	P3.0	P1.3	P1.2	P1.1	P1.0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

```
Dim we as byte, wy as byte, xx as byte
```

```
Do
```

```
    we = P3 AND &b00000011      ' maskowanie niewykorzystywanych bitów P3
    xx = P1 AND &b11110000      ' zapamiętanie 4 starszych bitów P1
    wy = lookup(we, Tablica)    ' odczytanie stanu wyjść
    P1 = xx OR wy               ' ustawienie 4 młodszych bitów P1
```

```
Loop
```

```
Tablica:
```

```
Data &b00000001 , &b00000010, &b00000100 , &b00001000
```



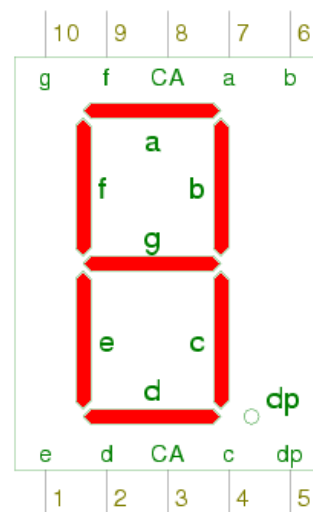
Napisz program realizujący programowo układ dekodera opisanego tabelą:

Wejścia		Wyjścia			
P3.1	P3.0	P1.3	P1.2	P1.1	P1.0
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	0
1	1	1	1	1	0



Przeanalizuj działanie transkodera BCD na kod wyświetlacza siedmiosegmentowego (UCY 7447). Uzupełnij poniższą tabelę.

Wejście KOD	Wyjścia								LED
	P1.0 a	P1.1 b	P1.2 c	P1.3 d	P1.4 e	P1.5 f	P1.6 g	P1.7 dp	
0									0
1									1
2									2
3									3
4									4
5									5
6									6
7									7
8									8
9									9
10									A
11									b
12									C
13									d
14									E
15									F



Napisz program realizujący programowo układ transkodera kodu binarnego na kod wyświetlacza 7 segmentowego ze wspólną anodą opisanego poniższą tabelą prawdy. Wejściem będzie wartość zmiennej KOD.



.....

.....

.....

.....

.....

2.7 Drivery - sterowanie silników krokowych

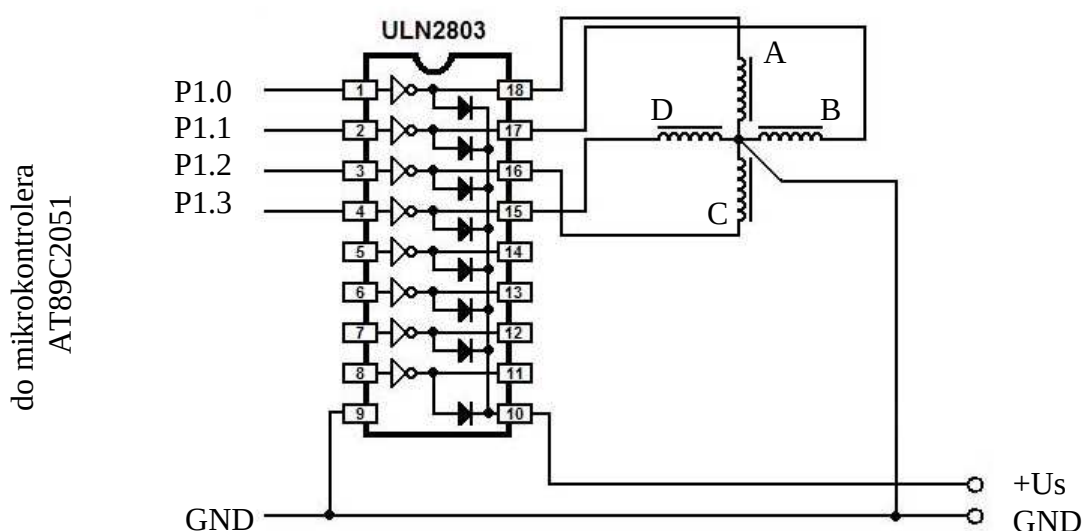
Data:

Ocena:

W trakcie realizacji ćwiczenia należy napisać i uruchomić programy sterowania pracą unipolarnego silnika krokowego oraz sformułować wnioski przez odpowiednie pomiary i obserwacje.

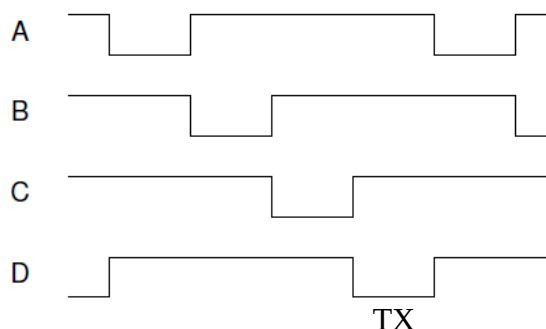


Schemat układu pomiarowego



Unipolarny silnik krokowy posiada wyprowadzone na zewnątrz 5 zacisków. Jeden z nich jest zaciskiem wspólnym dla wszystkich uzwojeń, podłączanym zazwyczaj do dodatniego bieguna zasilania (+Us). Pozostałe 4 zaciski to wyprowadzenia odpowiednich uzwojeń. Silnik krokowy jest podłączony do 4 najmłodszych bitów portu P1 mikrokontrolera przez wzmacniacze (drivery) układu ULN2803. Przepływ prądu przez wybraną cewkę silnika uzyskuje się podając wartość "0" na odpowiednie wyprowadzenie mikrokontrolera, do którego podłączona jest cewka. Na początku programu należy wszystkie wyjścia sterujące cewkami ustawić w stan "1".

Najprostszą metodą sterowania silnika krokowego jest sterowanie jednofazowe, w którym dla danej chwili czasu prąd przepływa tylko przez jedno uzwojenie silnika. Przebieg sygnałów sterujących dla sterowania jednofazowego przedstawia rysunek poniżej.





Dokończ program sterowania jednofazowego: $A \rightarrow B \rightarrow C \rightarrow D$. Uzupełnij program o odpowiednie komentarze.

' Przyporządkowanie nazw symbolicznych dla poszczególnych linii wyjściowych

```
_A Alias P1.0
_B Alias P1.1
_C Alias P1.2
_D Alias P1.3
TX Alias 10
```

P1=255 ' wszystkie wyjścia P1 w stanie "1"

Do

```
_A = 0
Waitms TX
_A = 1

_B = 0
Waitms TX
_B = 1
```

Loop



Zmianę kierunku obrotu uzyskuje się przez zmianę kolejności załączania cewek: $D \rightarrow C \rightarrow B \rightarrow A$ lub przez odpowiednią zmianę ich podłączenia.



Określ eksperymentalnie minimalne opóźnienie, dla którego silnik pracuje poprawnie (nie „gubi” kroków)

TX =



Jaki wpływ na pracę silnika ma zmiana opóźnienia T0?

.....

.....

.....



Ile kroków musi wykonać silnik żeby jego oś obróciła się o 360 stopni?

.....

.....

.....



Napisz program sterowania dwufazowego. Kolejność załączania faz: **AB → BC → CD → DA.**



Napisz program sterowania półkrokowego. Kolejność załączania faz: **A → AB → B → BC → C → CD → D → DA → A.**



.....

.....

.....

.....

.....

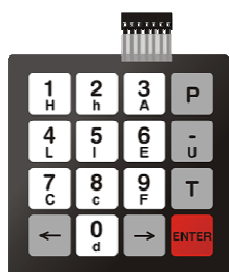
3 Komunikacja z użytkownikiem

Mikrokontrolery przetwarzają informację w postaci binarnej, która jest trudna do interpretacji przez użytkownika systemu, dlatego dane wejściowe i wyjściowe prezentuje się w bardziej dogodnych dla człowieka „kodach”. Dane wejściowe wprowadza się do układu mikroprocesorowego za pomocą różnego rodzaju klawiatur: opartych na przyciskach, matrycowych, membranowych czy też ekranów dotykowych. Bardzo dużą popularnością w układach cyfrowych cieszą się impulsatory umożliwiające szybką zmianę nastawy przypominającą działanie potencjometru. Możliwe jest także przesyłanie danych za pomocą zdalnych klawiatur i manipulatorów (IR, RF). Dane wyjściowe prezentuje się za pomocą wskaźników LED, segmentowych i graficznych wyświetlaczy LED, alfanumerycznych i graficznych wyświetlaczy LCD.



Przykładowe urządzenia wejścia i wyjścia.

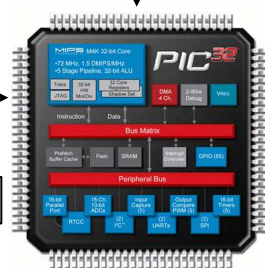
Klawiatura membranowa



Impulsator



Pilot RC5



Mikrokontroler



Wyświetlacze 7-segmentowe LED



Alfanumeryczny wyświetlacz LCD

3.1 Wyświetlacz alfanumeryczny LCD

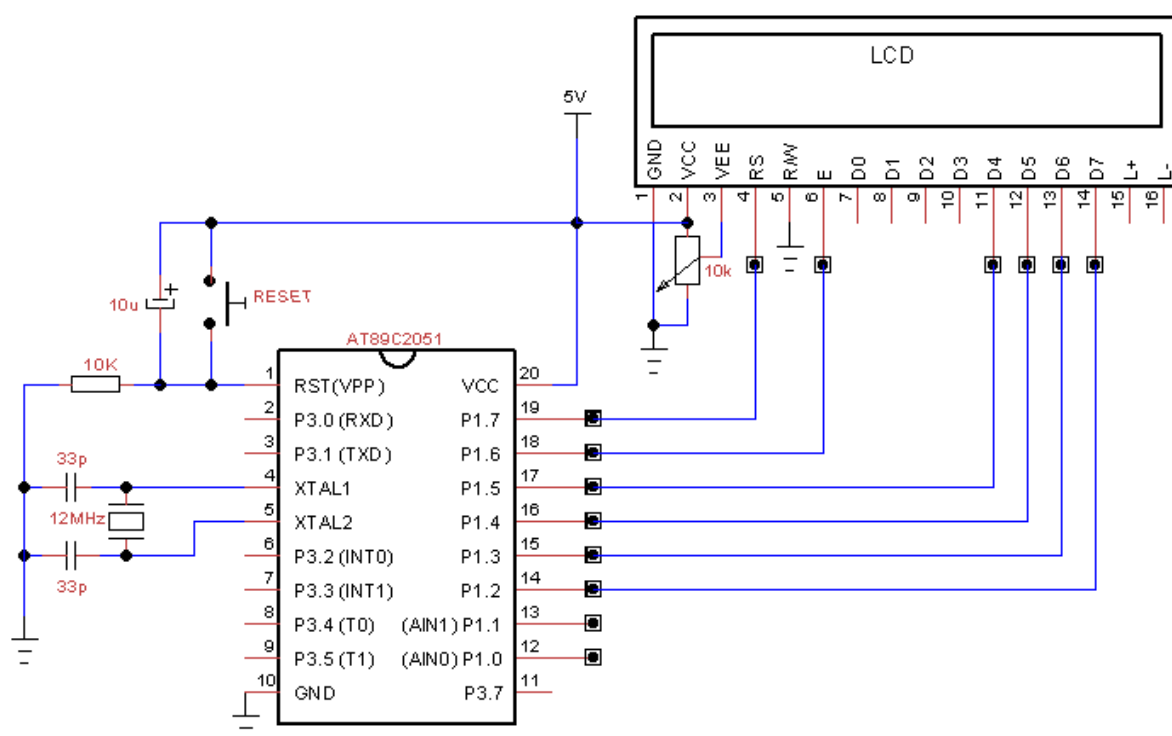
Data:

Ocena:

Alfanumeryczne wyświetlacze LCD ze sterownikiem HD44780 są jednymi z najczęściej stosowanych elementów w układach mikroprocesorowych. Ze względu na niską cenę oraz prostotę sterowania wyświetlacze oparte o kontroler HD44780 zyskały ogromną popularność. W ćwiczeniu przedstawione zostaną podstawowe informacje dotyczące obsługi wyświetlaczy ze sterownikiem HD44780 w środowisku BASCOM '51 DEMO.



Schemat układu pomiarowego (sterowanie 4-bitowe)

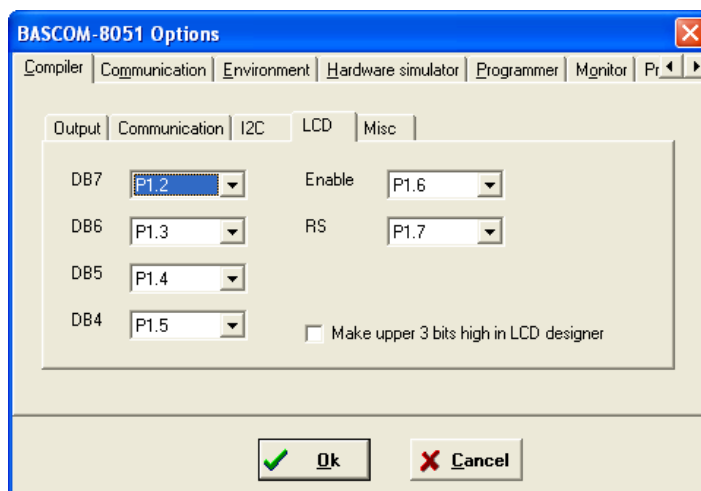


Funkcje wyprowadzeń wyświetlacza alfanumerycznego ze sterownikiem HD44780

Wyprowadzenie	Funkcja
GND	masa
VCC	+5V
VEE	kontrast
RS	„0” – rozkaz „1” - dana
R/W	„0” – zapis do LCD (dana lub rozkaz) „1” – odczyt z LCD (dana lub rozkaz)
E	„1” – zezwolenie na przepisanie informacji
D0..D7	Linie danych (kod znaku lub kod rozkazu)
L+, L-	Podświetlenie



konfiguracja wyświetlacza w środowisku BASCOM '51 DEMO (Options → Compiler → LCD)



Zestaw znaków, definiowanie własnych znaków

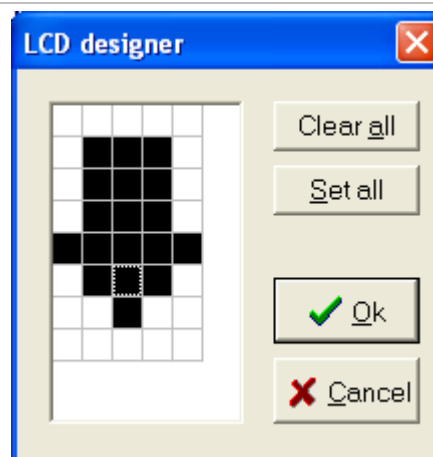
Standardowy zestaw znaków alfanumerycznego wyświetlacza HD44780

Char. code		0	0	0	0	0	0	1	1	1	1	1	1	1	1
		0	0	0	0	1	1	1	0	0	1	1	1	1	1
		0	0	1	1	0	0	1	1	1	0	0	1	1	1
		0	1	1	0	0	1	1	1	1	0	0	1	1	1
		0	0	1	0	1	0	1	0	1	0	1	0	1	0
xxxx0000		0	@	P	\	P	-	9	E	a	P				
xxxx0001		!	1	A	Q	a	q	.	7	4	ä	q			
xxxx0010		"	2	B	R	b	r	"	ı	ı	ı	ı	ı	ı	ı
xxxx0011		#	3	C	S	c	s	ı	ı	ı	ı	ı	ı	ı	ı
xxxx0100		\$	4	D	T	d	t	.	ı	ı	ı	ı	ı	ı	ı
xxxx0101		%	5	E	U	e	u	.	ı	ı	ı	ı	ı	ı	ı
xxxx0110		&	6	F	V	f	v	.	ı	ı	ı	ı	ı	ı	ı
xxxx0111		'	7	G	W	g	w	.	ı	ı	ı	ı	ı	ı	ı
xxxx1000		(	8	H	X	h	x	.	ı	ı	ı	ı	ı	ı	ı
xxxx1001		)	9	I	Y	i	y	.	ı	ı	ı	ı	ı	ı	ı
xxxx1010		*	:	J	Z	j	z	.	ı	ı	ı	ı	ı	ı	ı
xxxx1011		+	;	K	[	k	[	.	ı	ı	ı	ı	ı	ı	ı
xxxx1100		,	<	L	¥	l	ı	.	ı	ı	ı	ı	ı	ı	ı
xxxx1101		-	=	M	]m	.	ı	.	ı	ı	ı	ı	ı	ı	ı
xxxx1110		.	>	N	^n	.	ı	.	ı	ı	ı	ı	ı	ı	ı
xxxx1111		/	?	O	_o	.	ı	.	ı	ı	ı	ı	ı	ı	ı

Wyświetlenie znaku stopnia można zrealizować za pomocą wywołania:

Lcd Chr(&B11011111)

Definiowanie własnych znaków (Tools → LCD Designer)



Wynikiem działania kreatora własnych znaków jest dodanie na początku programu definicji znaku:

Deflcdchar ?,32,14,14,14,31,14,4,32

Gdzie znak ? zastępujemy kodem znaku (0-7)

Użycie znaku (kod=0):

Lcd Chr(0)



Podstawowe funkcje obsługi wyświetlacza LCD

Cls	' czyszczenie wyświetlacza
Locate 3,1	' ustawienie kursora w pozycji y=3, x=1
LCD „Pracownia”	' wyświetlenie napisu Pracownia
Shift lcd Right	' przesunięcie napisu o jeden znak w prawo
Home	' przesunięcie kursora do początku linii

Uwaga: Funkcja LCD pozwala na podanie kilku parametrów do wyświetlenia (łańcuch znaków, zmienna itp.). Kolejne parametry oddzielamy znakiem średnika.

Przykład: Wyświetlenie temperatury umieszczonej w zmiennej T (Chr(0) – np. znak stopnia).

Lcd „Temperatura=" ; T ; **Chr**(0) ; „C”



Napisz program wyświetlający Twoje imię i nazwisko w kolejnych liniach wyświetlacza. Następnie przesun napis o dwa znaki w prawo.



Napisz program wyświetlający zawartość zmiennej Licznik w postaci łańcucha Licznik=x, gdzie x oznacza zawartość zmiennej licznik. Zmienna powinna być inkrementowana w każdym kroku pętli. Opóźnienie zliczania powinno wynosić około 1 sekundy.



Napisz program wyświetlający Twoje imię i nazwisko w kolejnych liniach wyświetlacza. Napis powinien być drukowany litera po literze. Opóźnienie „drukowania” powinno wynosić około 0.2 sekundy.



.....

.....

.....

.....

.....

3.2 Wyświetlacze LED

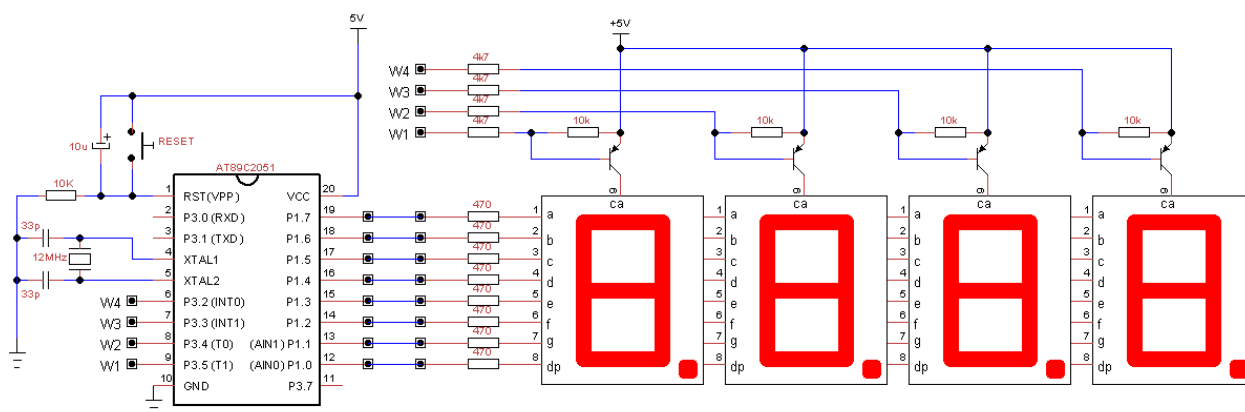
Data:

Ocena:

Wyświetlacze LED stosowane są wszędzie tam, gdzie wymagany jest duży kontrast i znaczna jasność świecenia. Najbardziej popularne są wyświetlacze 7-segmentowe pozwalające na wyświetlanie informacji cyfrowej. Zespoły wyświetlaczy można sterować niezależnie lub w sposób multipleksowany (sekwencyjny). Druga metoda jest częściej stosowana ze względu na znaczną oszczędność w liczbie sygnałów potrzebnych do ich wystawiania. Celem ćwiczenia jest zapoznanie się z metodami sterowania 7-segmentowymi modułami wyświetlaczy.

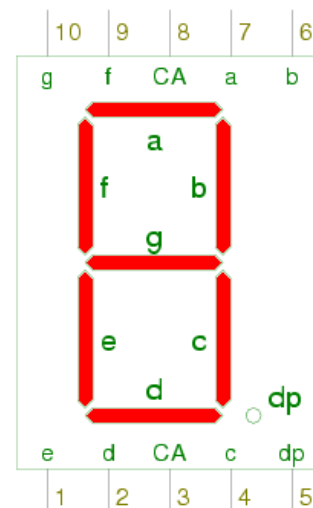


Schemat układu pomiarowego (sterowanie 4-bitowe)



Uzupełnij poniższą tabelę oraz na jej podstawie definicję tablicy kodów LED7SEG:

Wejście KOD	Wyjścia								LED
	P1.0 dp	P1.1 g	P1.2 f	P1.3 e	P1.4 d	P1.5 c	P1.6 b	P1.7 a	
0	1	1	0	0	0	0	0	0	0
1									1
2									2
3									3
4									4
5									5
6									6
7									7
8									8
9									9
10									A
11									b
12									C
13									d
14									E
15									F





Dokończ program sterowania sekwencyjnego (wyświetl liczbę 1234):

```
P3 = P3 OR 15      ' zgaszenie wszystkich wyświetlaczy
Do
  P1 = lookup(1, LED7SEG)  ' P1 = kod cyfry „1”
  P3.5 = 0                ' włączenie pierwszego wyświetlacza
  waitms 1                ' wyświetlacz włączony przez 1ms
  P3.5 = 1                ' zgaszenie wyświetlacza
```

Loop

```
LED7SEG:
Data &b11000000,
```



Oblicz częstotliwość wyświetlania dla programu z poprzedniego ćwiczenia.

f =



Zmieniaj opóźnienie (czas włączenia wyświetlacza) i sformułuj odpowiednie wnioski z tych obserwacji. Przy jakim opóźnieniu nie widać efektu migotania?

.....

.....

.....



Dokończ program konwersji liczby całkowitej umieszczonej w zmiennej X na kody wskaźników 7-segmentowych sterowanych sekwencyjnie.

```
Dim X As Integer, kod As byte
P3 = P3 OR 15          ' zgaszenie wszystkich wyświetlaczy
X = 4321               ' ustalenie zmiennej
Do
    kod = X Mod 10     ' kod jednostek
    X = X/10           ' X = X/10
    P1 = lookup(kod, LED7SEG) ' P1 = kod jednostek
    P3.2 = 0           ' włączenie wyświetlacza jednostek
    Waitms 1           ' wyświetlacz włączony przez 1ms
    P3.2 = 1           ' zgaszenie wyświetlacza
```

```
    X = X+1            ' zmiana X
Loop
```

```
LED7SEG:
Data &b11000000,
```



.....

.....

.....

.....

.....

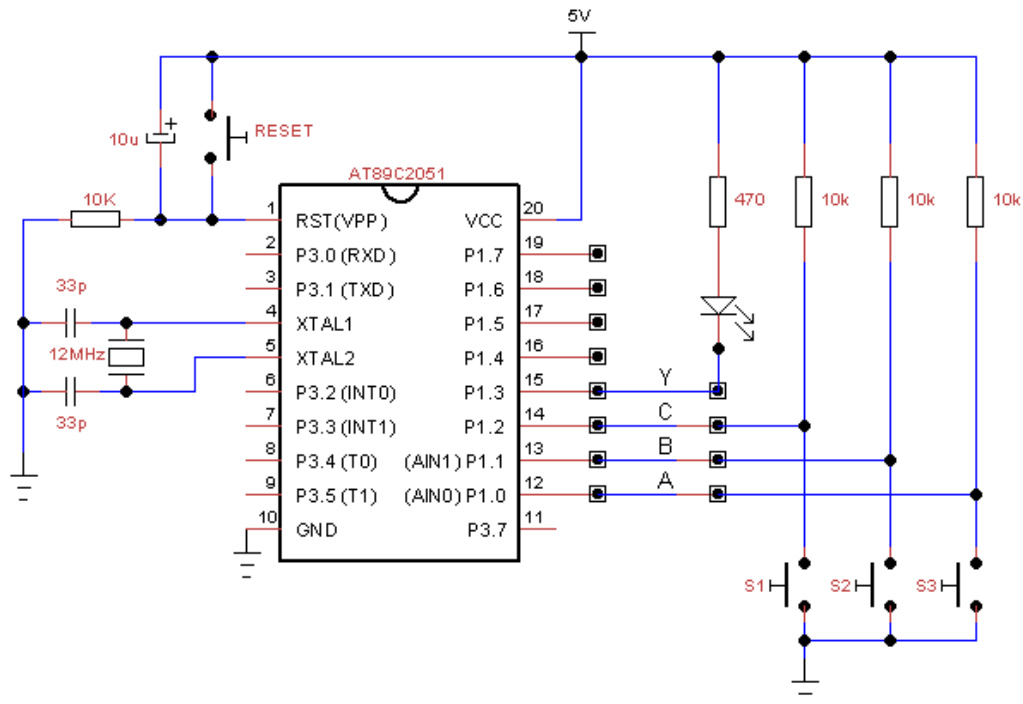
3.3 Obsługa klawiatur

Data:

Ocena:



Schemat układu pomiarowego (sterowanie prostej klawiatury)



Dokończ program obsługi klawiatury trzyprzyciskowej. Program ma realizować trzy funkcje P=1: dioda wyłączona, P=1: dioda włączona, P=2: dioda „mruga” z częstotliwością 1Hz.

```

Dim P As byte
P1 = 255
P=1
Do
    If P1.2=0 Then P=1
        ' zmienna P zapamiętuje realizowany program
        ' zgaszenie diody i ustawienie wejść
        ' ustalenie realizowanej funkcji

        ' sprawdzenie wciśnięcia klawisza S1

        If P=1 Then P1.3=1
        If P=2 Then
        If P=3 Then

        ' P=1: dioda wyłączona

    End if
Loop

```



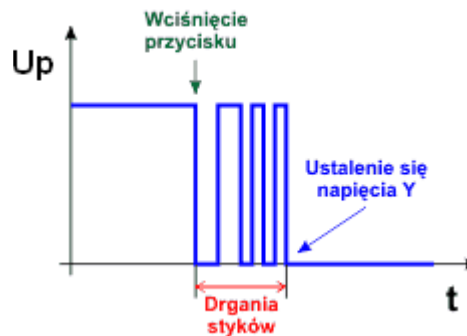

Zmodyfikuj część programu z poprzedniego ćwiczenia odpowiedzialną za odczytywanie stanu wciśnięcia klawiszy w taki sposób, żeby zmiana funkcji realizowana była za pomocą jednego przycisku. Program ma realizować trzy funkcje P=1: dioda wyłączona, P=1: dioda włączona, P=2: dioda „mruga” z częstotliwością 1Hz.

```
If P1.2=0 Then ' sprawdzenie wciśnięcia klawisza S1
```

```
End if
```



Zmodyfikuj program eliminując wpływ drgań zestyków (podpowiedź – zastosuj niewielkie opóźnienie pomiędzy kolejnymi sprawdzeniami stanu przycisków)



.....

.....

.....

.....

.....

.....

3.4 Transmisja szeregową - UART, RS232

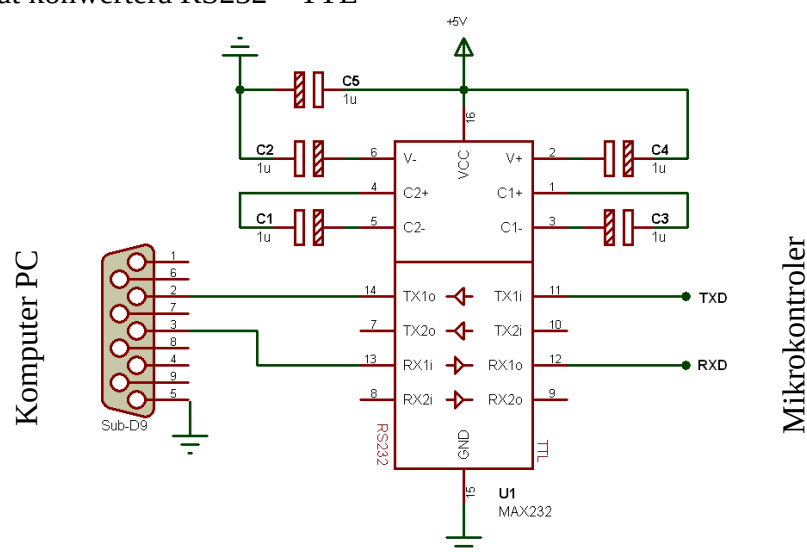
Data:

Ocena:

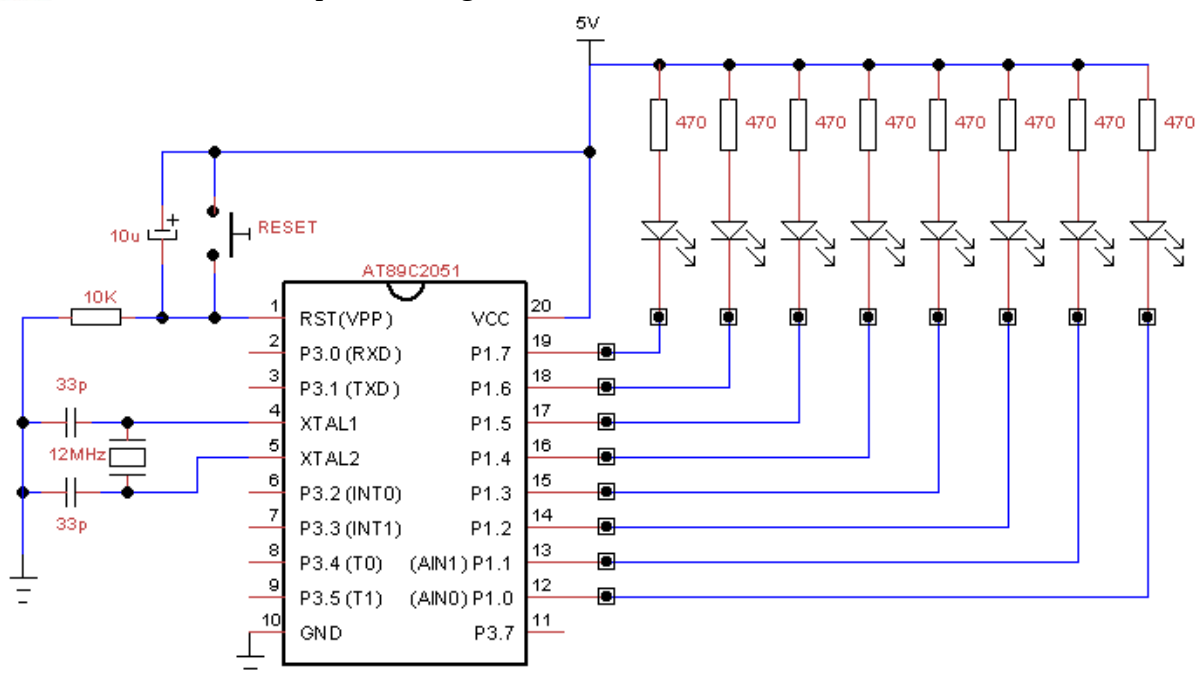
Mikrokontrolery '51 posiadają zintegrowany moduł UART (ang. *Universal Asynchronous Receiver and Transmitter*) służący do asynchronicznego przekazywania i odbierania informacji poprzez port szeregowy. Łączenie mikrokontrolera z komputerem PC wymaga użycia konwertera np. MAX232 służącego do dopasowania sygnałów interfejsu RS232C, do interfejsu spotykanego w mikrokontrolerach, gdzie poziomy napięcia to 0-5V. Porty komputera oznaczane są odpowiednio: COM1-COM4.



Schemat konwertera RS232 – TTL



Schemat układu pomiarowego





Programowanie interfejsu szeregowego w BASCOM.

```

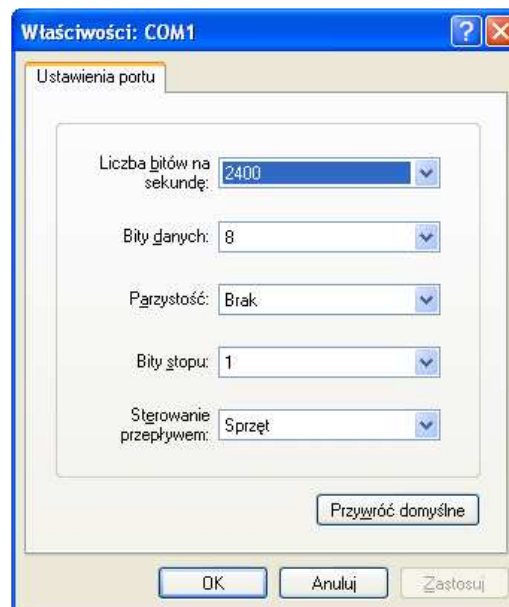
$baud      = 2400      ' szybkość UART=2400 bodów
$crystal    = 12000000 ' XTAL=12MHz

Dim x As Byte
Do
  Print "Wprowadź znak: " ' zapis do UART
  Input x                  ' odczyt z UART
  P1 = x                  ' Zapal diody - kod ASCII!!!
  Inputhex "Wprowadź liczbę: " , x ' odczyt z UART liczby HEX
  P1 = x                  ' Zapal diody - kod HEX
Loop

```



Konfiguracja połączenia w programie Hyper Terminal (Start → Akcesoria → Komunikacja → Hyper Terminal)



.....

.....

.....

.....

.....

4 Układy czasowe, przerwania

Mikrokontrolery rodziny '51 posiadają dwa 16-bitowe liczniki, które mogą być konfigurowane niezależnie jako liczniki impulsów wewnętrznych (czasomierz - Timer) lub zewnętrznych (licznik impulsów zewnętrznych - Counter).

Podczas pracy w funkcji czasomierza, pracujące w danym trybie rejestry licznika są inkrementowane z każdym cyklem maszynowym. Cykl maszynowy w klasycznym mikrokontrolerze rodziny '51 trwa 12 okresów sygnału taktującego mikrokontroler, zatem częstotliwość zliczania jest równa 1/12 częstotliwości sygnału taktującego lub inaczej, licznik jest inkrementowany co czas równy 12 okresom sygnału taktującego.

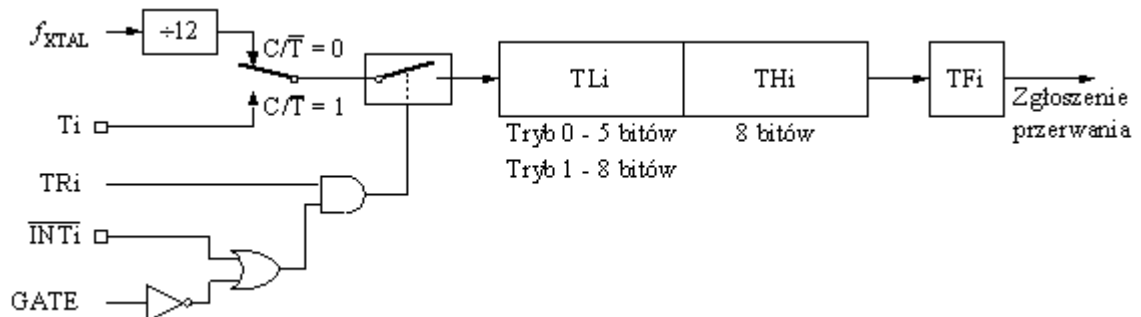
Przy konfiguracji w funkcji licznika, pracujące rejestry są inkrementowane przy każdym zboczu opadającym sygnału doprowadzonego do odpowiadającego dla danego licznika wyprowadzenia zewnętrznego T0 (P3.4) lub T1 (P3.5) mikrokontrolera.

Ze względu na fakt, iż wykrycie zmiany z 1 na 0 na wejściu T0 lub T1 wymaga dwóch cykli maszynowych (24 okresów oscylatora), to w przypadku pracy w funkcji licznika impulsów zewnętrznych maksymalna częstotliwość zmian możliwa do poprawnego wykrycia jest 24-krotnie mniejsza od częstotliwości dołączonego do mikrokontrolera oscylatora.

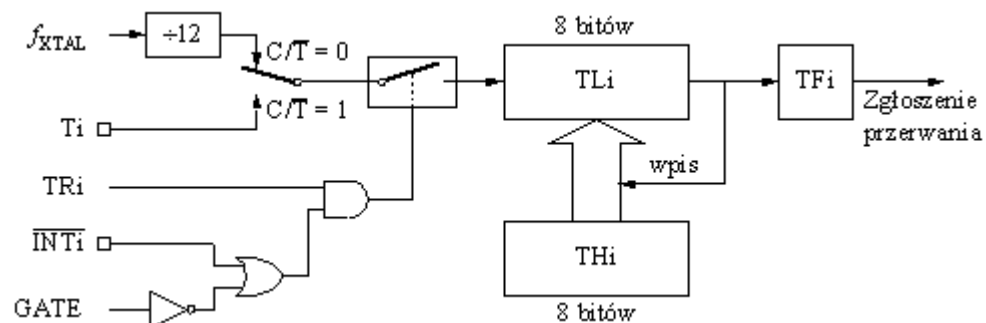
Zawartość liczników zapamiętywana jest w rejestrach TLi, THi, gdzie i oznacza numer licznika 0 lub 1.

Liczniki mogą pracować w 4 trybach (najczęściej wykorzystuje się tryby 1 i 2):

- **Tryb 0** – Rejestr THi pracuje jako 8-bitowy licznik taktowany przez 5-bitowy dzielnik TLi, dając w sumie licznik 13-bitowy.



- **Tryb 1** – Rejestry THi i TLi pracują jako jeden licznik 16-bitowy.
- **Tryb 2** – Rejestr TLi pracuje jako 8-bitowy licznik z automatycznym przeładowaniem wartości początkowej z rejestru THi.



- **Tryb 3** – Licznik T0 pracuje jako dwa niezależne 8-bitowe liczniki: TL0 sterowany za pomocą bitów sterujących licznika T0, TH0 sterowany za pomocą bitów sterujących licznika T1. Licznik T1 – zatrzymany.

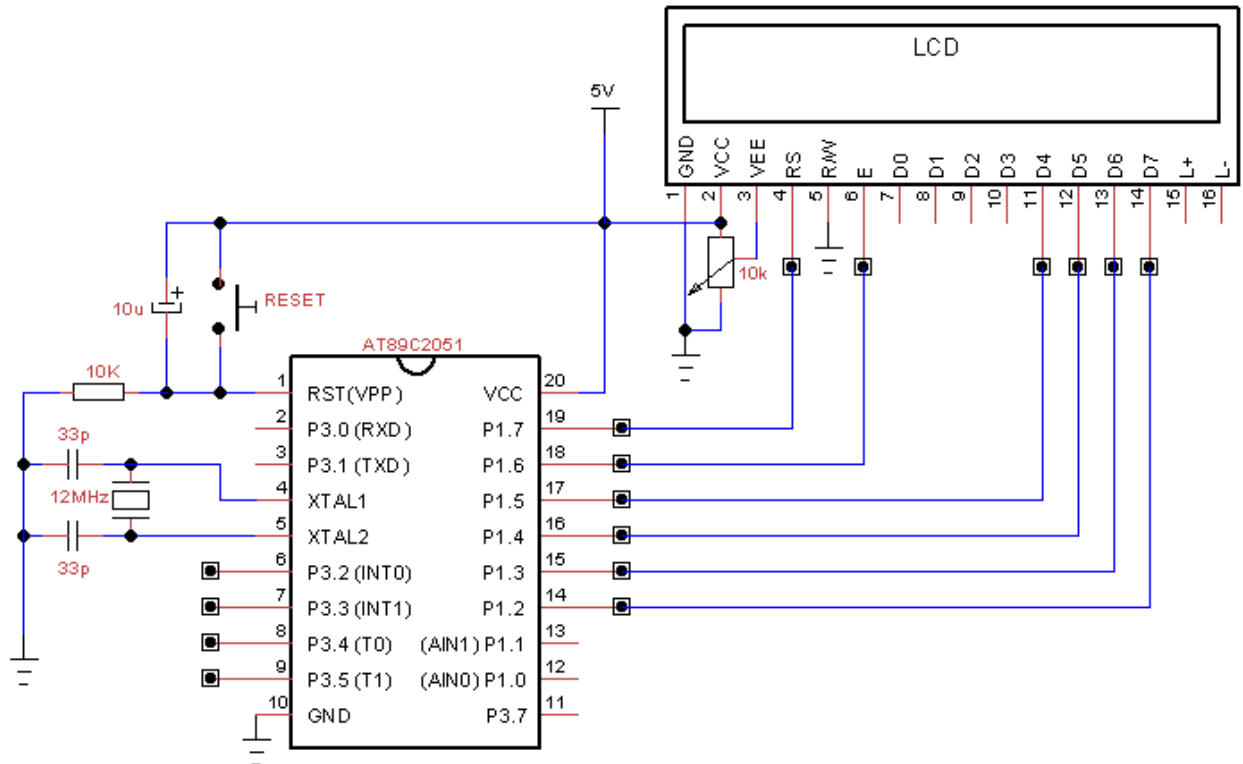
4.1 Licznik zdarzeń

Data:

Ocena:



Schemat układu pomiarowego



Uruchom program prostego licznika zdarzeń. Dołącz do wejścia T0 przycisk i sprawdź reakcję programu na jego wciskanie. Następnie podłącz generator TTL i obserwuj działanie programu dla różnych częstotliwości sygnału.

'Timer0 jako licznik 16 bitowy, bez bramkowania.

Config Timer0 = Counter , Mode = 1 , Gate = Internal

Counter0 = 0 ' Wartość początkowa licznika.

Start Counter0 ' Włączenie licznika

Do

 Cls

 Home

 LCD "Licznik=", Counter0

Loop

End



Uruchom program prostego licznika zdarzeń opartego na zliczaniu przerwań zewnętrznych ze źródła INT0 (P3.2). Dołącz do wejścia INT0 przycisk i sprawdź reakcję programu na jego wciskanie. Następnie podłącz generator TTL i obserwuj działanie programu dla różnych częstotliwości sygnału.

```
On Int0 Przerwanie0      'skok do Przerwanie0 gdy wystąpi przerwanie
Enable Interrupts
Enable Int0              'włączenie przerwania od INT0
Dim L As Long
L = 0

Do
  Cls
  Home
  LCD "Licznik=", L
Loop

Przerwanie0:
  L = L+1
Return
```



Przy jakiej wartości nastąpi przepełnienie licznika?

.....

.....

.....



Czy w tego typu obsłudze zliczania impulsów można programowo wyeliminować drgania zestyków?

.....

.....

.....



.....

.....

.....

.....

.....

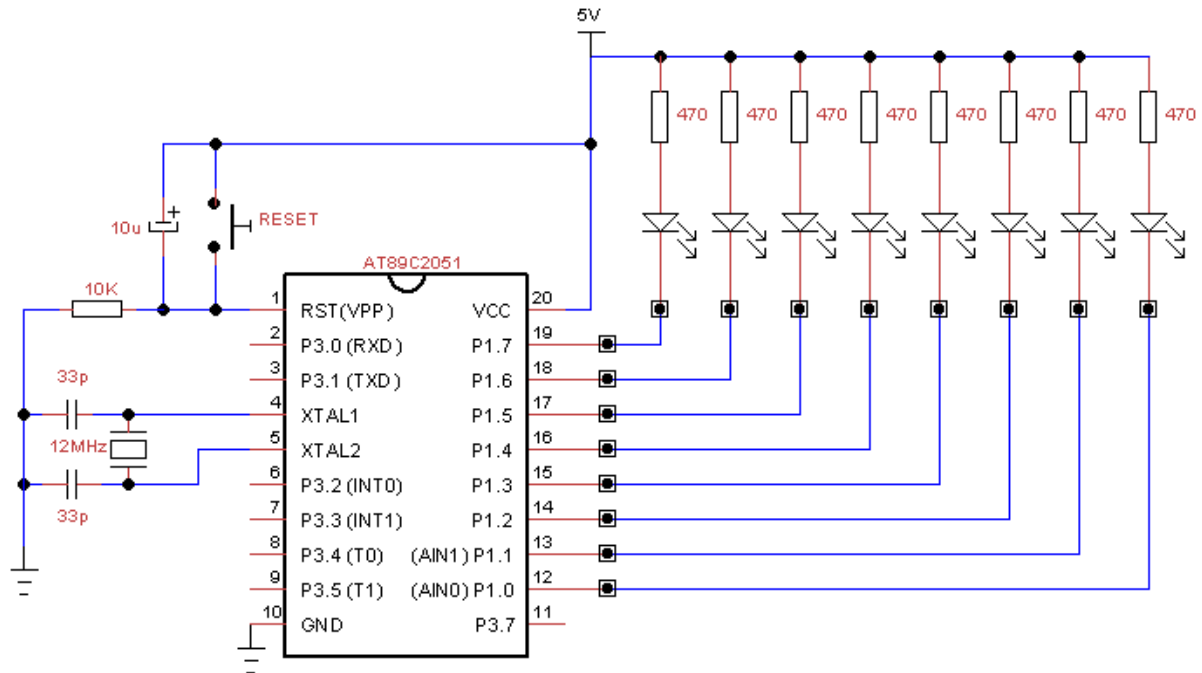
4.2 Precyzyjny generator

Data:

Ocena:



Schemat układu pomiarowego



Uruchom program prostego generatora sygnału prostokątnego. Wyjściem generatora niech będzie linia P1.0. Dla małych częstotliwości obserwuj sygnał świecenia diody, przy większych obserwuj przebieg na wyjściu za pomocą oscyloskopu.

```
'Timer0 jako 8-bitowy czasomierz, tryb 2, bez bramkowania.
Config Timer0 = Timer , Gate = Internal , Mode = 2
On Timer0 Przepelnienie      'podprogram obsługi przerwania Przepelnienie
Load Timer0, 250             'przepełnienie co 250us dla XTAL=12MHz
Enable Timer0                 'włączenie przerwań od Timer0
Enable Interrupts             'włączenie przerwań globalnie
Start Timer0                  'włączenie Timera
Dim L As Word
L = 0

Do                               'pusta pętla
Loop

'podprogram obsługi przerwania - przepełnienie Timera0
Przepełnienie:
    L=L+1
    If L=400 Then               'zmiana stanu P1.3 co 250us*400 = 100ms
        P1.0 = Not P1.0
        L=0
    Endif
Return
```



Narysuj przebieg uzyskany na wyjściu P1.0. Oblicz częstotliwość przebiegu.

f =



Zmodyfikuj program w celu uzyskania częstotliwości przebiegu równej 1kHz.
Przedstaw zmodyfikowane fragmenty kodu



Uruchom i sprawdź działanie programowego modulatora szerokości impulsów PWM (ang. Pulse Width Modulation).

'podprogram obsługi przerwania - przepełnienie Timera0

Przepełnienie:

L=L+1

If L<10 Then P1.1=1 Else P1.1=0

If L=40 Then L=0

' współczynnik wypełnienia: 10/40

' okres PWM: 250us*40 = 10ms

Return



Jaki wpływ na świecenie diody ma zmiana okresu PWM a jaki współczynnika wypełnienia?

.....

.....

.....



.....

.....

.....

.....

.....

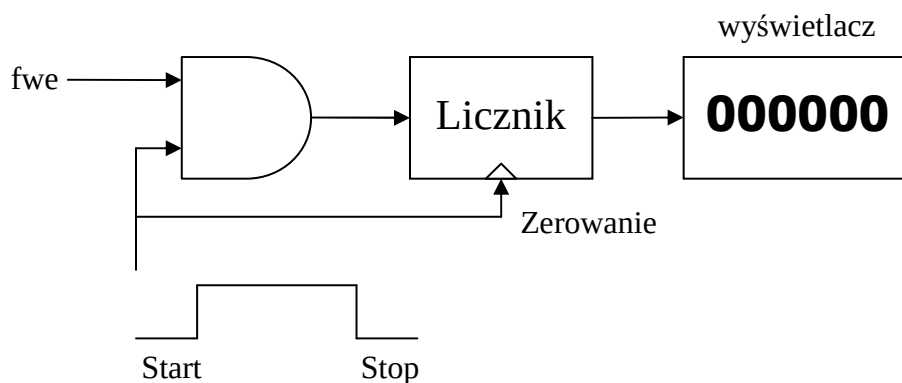
4.3 Pomiar częstotliwości

Data:

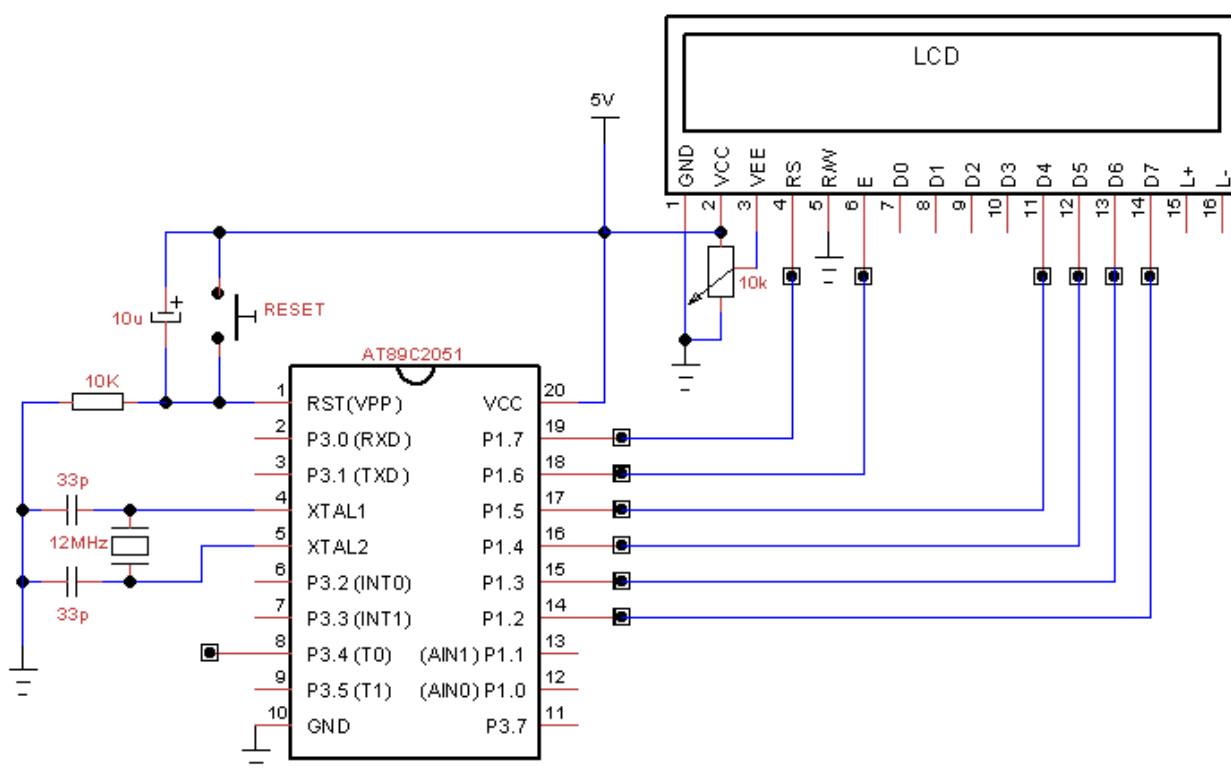
Ocena:



Pomiar częstotliwości polega na zliczeniu okresów sygnału wejściowego w dokładnym przedziale czasu (tzw. czasie bramkowania). Przy czasie bramkowania równym jednej sekundzie wynik zliczania określa częstotliwość w Hz. Na początku pomiaru układ powinien automatycznie wyzerować licznik.



Schemat układu pomiarowego





Uruchom program prostego miernika częstotliwości sygnału TTL. Na wejście licznika T1 (P3.4) podaj sygnał z generatora TTL o znanej częstotliwości.

```
'Timer1 jako 16-bitowy licznik, tryb 1, bez bramkowania.
Config Timer0 = Counter , Gate = Internal , Mode = 1
On Timer0 Przepelnienie1
Enable Interrupts           'włączenie przerwań globalnie
Dim L As Word, F As Long, Ftmp As Long, T0cnt As Long
Do
    Counter0 = 0
    Enable Counter0         'włączenie przerwań od Timer0
    Start Counter0

    Wait 1                   'czas pomiaru ok. 1s!!!

    Stop Counter0
    Disable Counter0        'wyłączenie przerwań od Timer0
    Ftmp = Counter0         'obliczenie wyniku pomiaru
    F = T0cnt*65536
    F = F + Ftmp
    Cls
    Home
    Lcd "F="; F ; "[Hz]"
Loop

'podprogram obsługi przerwania - przepełnienie Timera1 co 65536 impulsów
Przepełnienie1:
    T0cnt = T0cnt+1
Return
```



Uzupełnij tabelę.

fg (wzorzec)	f (zmierzona)	$\Delta f = f - fg$ (błąd bezwzględny)	$\delta f = \frac{\Delta f}{f_w} \cdot 100\%$
100Hz			
1kHz			
10KHz			
100KHz			



Zamień instrukcję Wait 1 na kod podany poniżej. Dobierz eksperymentalnie wartość końcową licznika pętli For celem uzyskania dokładniejszego opóźnienia 1s.

```
For L=0 To 50000           'zmienić liczbę 5000 dla T pomiaru ~ 1s
Next L
```



.....

.....

.....

.....

.....

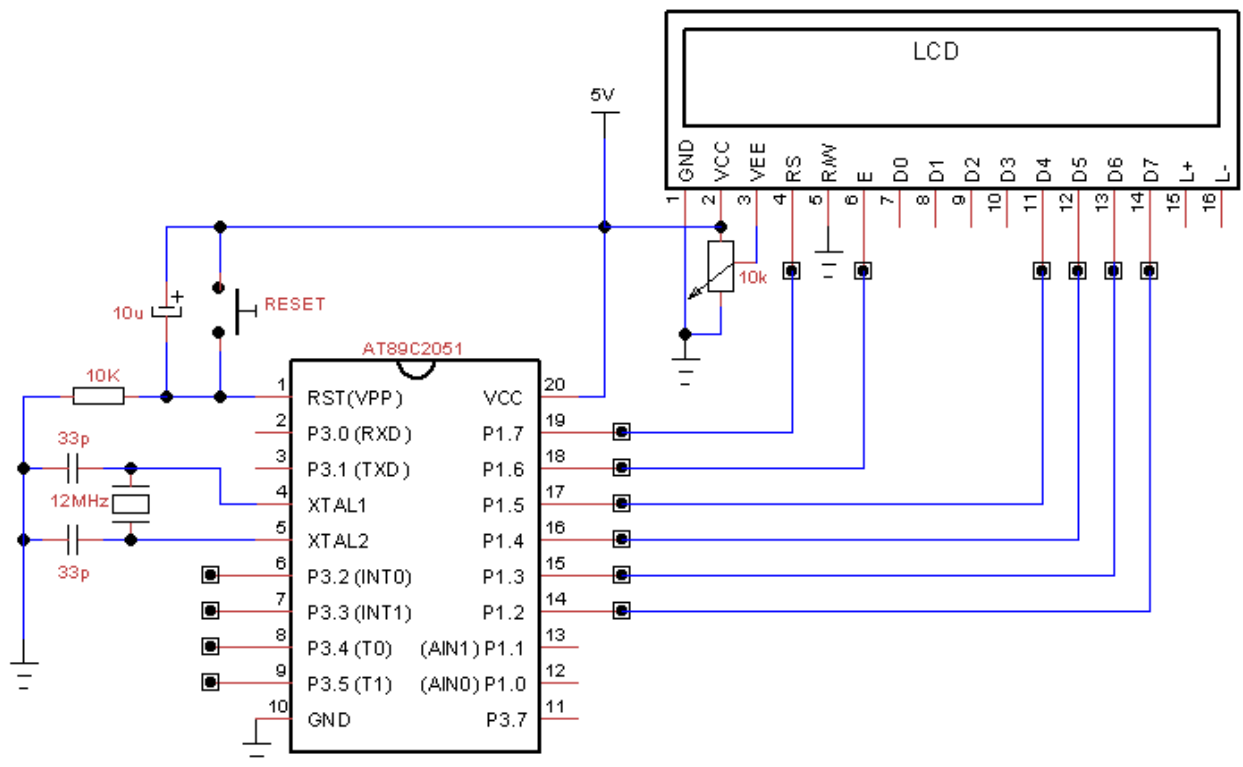
4.4 Zegar cyfrowy

Data:

Ocena:



Schemat układu pomiarowego



Uruchom program bardzo prostego zegarka cyfrowego.

```

Dim G As Byte , M As Byte , S As Byte
Cls
Do
  If S = 60 Then
    M = M + 1
    S = 0
  End If
  If M = 60 Then
    G = G + 1
    M = 0
  End If
  If G = 24 Then
    G = 0
  End If
  Cls
  Home
  LCD BCD(G); ":"; BCD(M); ":"; BCD(S)
  Wait 1
  S=S+1
Loop

```



Sprawdź działanie programu po zmianie kodu odpowiedzialnego za wyświetlanie czasu..

Zastąp:

```
Cls
Home
LCD G; ":"; M; ":"; S
```

przez:

```
Cls
Do
...

Locate 1,1
If G<10 Then LCD "0"
LCD G; ":"
If M<10 Then LCD "0"
LCD M; ":";
If S<10 Then LCD "0"
LCD S
```



Jaki wpływ na wyświetlanie czasu miała powyższa zmiana?

.....

.....

.....



W jaki sposób poprawić dokładność zegarka?

.....

.....

.....



Zmodyfikuj mechanizm zmiany zmiennej S za pomocą przerwania od układu czasowego Timer0.

```

Dim Count As Integer
Config Timer0 = Timer , Gate = Internal , Mode = 2
On Timer0 Timer_0_int
Load Timer0 , 250
Enable Interrupts
Enable Timer0
Start Timer0

Do
    If S = 60 Then
        M = M + 1
        S = 0
    End If
    If M = 60 Then
        G = G + 1
        M = 0
    End If
    If G = 24 Then
        G = 0
    End If
    ...
    Waitms 100
Loop

' podprogram obsługi przerwania Timera0
Timer_0_int:
    Count = Count + 1
    If Count = 4000 Then ' co 4000 przepełnień Timer0 zmodyfikuj S
        S = S + 1
        Count = 0
    End If
Return

```



Wyznacz okres przepełnienia czasomierza Timer0 oraz okres zmiany zmiennej S w procedurze obsługi przerwania Timer_0_int. Załóż $F_{XTAL}=12\text{MHz}$.

$$T_{Timer_0_int} = T_{XTAL} \cdot 250 = \dots\dots\dots$$

$$T_{S=S+1} = T_{Timer_0_int} \cdot 4000 = \dots\dots\dots$$



.....

.....

.....

.....

.....

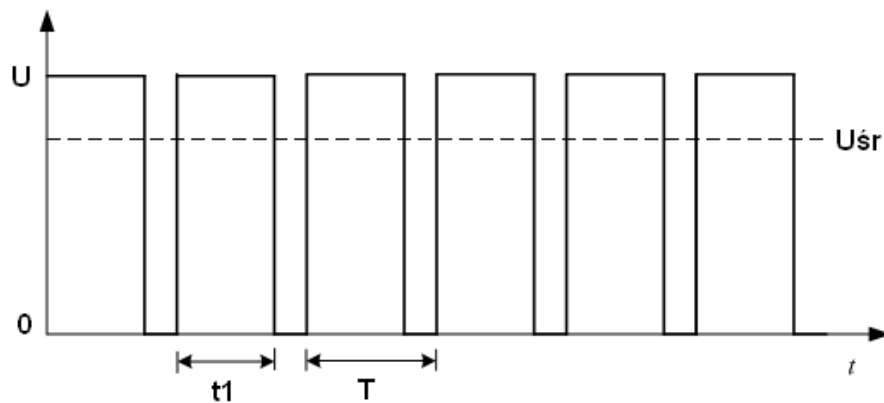
4.5 PWM

Data:

Ocena:

i

Modulacja szerokości impulsu PWM (ang *Pulse Width Modulation*) ma zastosowanie w wielu dziedzinach nauki i techniki. Najbardziej powiązana jest z elektroniką, telekomunikacją i automatyką. Stosuje się ją do sterowania sygnału prądowego lub napięciowego. Jest używana do wysterowania tranzystorów mocy w zasilaczach impulsowych, układach sterujących pracą silników prądu stałego, wzmacniaczach cyfrowych, regulatorach oświetlenia, temperatury itp. Zastosowanie tego typu modulacji zmniejsza znacznie straty mocy w układach sterujących zapewniając jednocześnie dużą dokładność sterowania urządzeniem. Zastosowaniu filtra wygładzającego umożliwia realizację prostego przetwornika C/A.



gdzie:

t1 – czas załączenia, T – okres

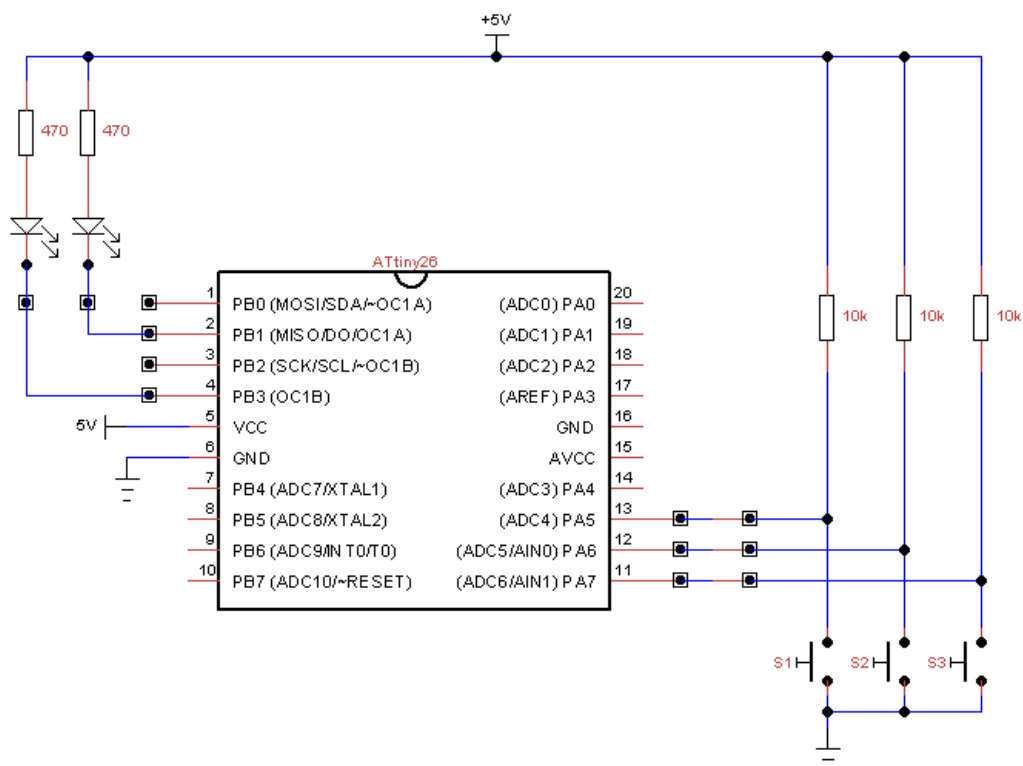
 $\tau = \frac{t1}{T}$ - współczynnik wypełnienia

Wartość średnia sygnału jest proporcjonalna do współczynnika wypełnienia:

$$U_{sr} = U \cdot \tau$$



Schemat układu pomiarowego



Uruchom program testujący działanie licznika Timer1 mikrokontrolera ATtiny26 w trybie PWM.

```

Config Portb = &B000001010 'Portb.1 (OC1A) Portb.3 (OC1B) pracują jako wyjście
Set Portb.1 'włączenie rezystora podciągającego
Set Portb.4 'włączenie rezystora podciągającego

'Timer0 jako PWM, Preskaler=1, Okres PWM=256*Preskaler/XTAL
Config Timer1 = Pwm , Compare A Pwm = Clear Down , Prescale = 1
Tccr1a = &B100000011
Tccr1b = &B000000001
Ocr1c = 255
Ocr1a = 64 'współczynnik wypełnienia 64/256 = 25%
Ocr1b = 25 'współczynnik wypełnienia 25/256 ~ 10%

```



Oblicz okres sygnału z powyższego programu.

$$T_{PWM} = T_{OSC} \cdot Ocr1c \cdot Prescale$$

T_{PWM} =



Zmodyfikuj program w taki sposób aby można było za pomocą przycisków zmieniać współczynnik wypełnienia: S1 – zwiększanie, S2 - zmniejszanie.



.....

.....

.....

.....

.....

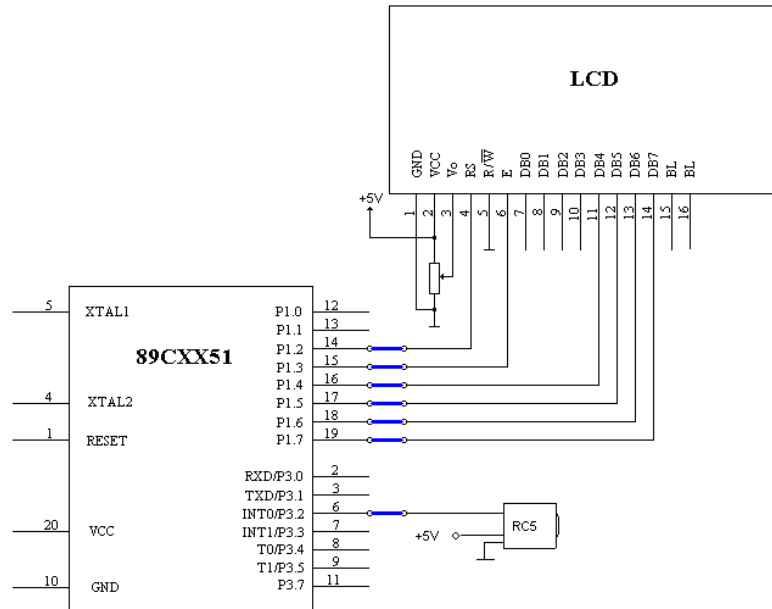
4.6 RC5

Data:

Ocena:



Schemat układu pomiarowego



i

Procedura GETRC5 umożliwiającą odbiór danych z odbiornika RC5 jest „fabrycznie” przypisana do wejścia INT0 procesora i **nie może** współpracować jakimkolwiek innym wejściem, nawet jeżeli jest to wejście przerwania sprzętowego. Dekoder podczerwieni TFMS lub SFH **musi** być dołączony do nóżki procesora będącej wejściem przerwania INT0, czyli w przypadku układu 2051 do pinu 6.



Uruchom program testujący działanie odbiornika RC5.

```
Dim Kod As Bit
Dim Command As Byte , Subaddress As Byte
$crystal = 12000000

Reset Tcon . 0
On Int0 Receiverc5
Enable Int0      'wyzwalanie narastajacym zboczem
Enable Interrupts

Cls
Cursor Off
Lcd "Waiting for RC5"      'odebrano nowy kod

Do
  If Kod = 1 Then
    Disable Int0
    Cls
    Lcd "Com:" ; Command ; ",Adr:" ; Subaddress
    Kod = 0      'zeruj flage
    Enable Int0
  End If
Loop

Receiverc5:
  Getrc5 ( Subaddress , Command )
  Kod = 1
Return
```



.....

.....

.....

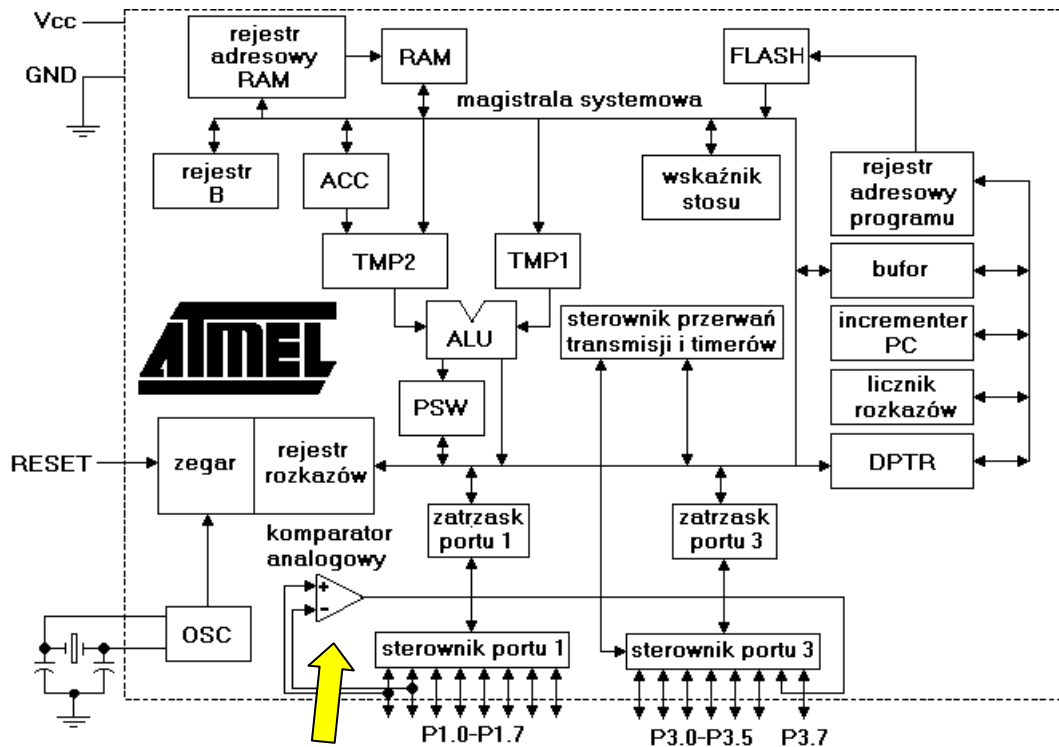
.....

.....

.....

5 Komparator analogowy

Mikrokontrolery AT89Cxx51 posiadają komparator analogowy podłączony do linii P1.0 (wejście nieodwracające – AIN0), P1.1 (wejście odwracające – AIN1) oraz wewnętrznie sprzęgnięte z linią P3.6 wyjście.



Komparator analogowy można wykorzystać w układach regulatorów, gdzie wymagane jest porównanie sygnału zmierzonego z napięciem wzorcowym. Są one najprostszym blokiem pozwalającym na dokonanie 1-bitowej konwersji sygnału z postaci analogowej na cyfrową. Często w tych układach można ustawić próg komparacji (poziom komparacji) oraz warunek komparacji, tzn. co się stanie w przypadku, gdy napięcie wejściowe wzrośnie powyżej progu komparacji (gdy ustawiliśmy taki warunek) lub zmaleje poniżej jego. Najczęściej ustawiana jest odpowiednia flaga i generowane jest przerwanie, o ile nie zostało zablokowane. Stosując jeden dodatkowy kondensator i rezystor (lub źródło prądowe) można za pomocą komparatora zbudować prosty przetwornik analogowo-cyfrowy.

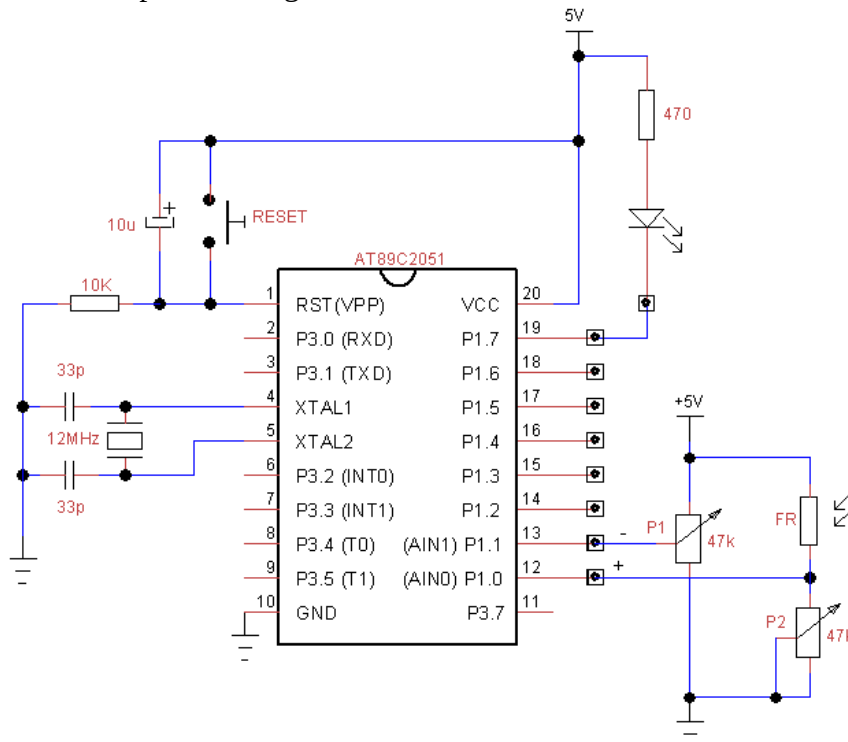
5.1 Komparator analogowy – prosty regulator oświetlenia

Data:

Ocena:



Schemat układu pomiarowego



Uruchom program prostego wyłącznika zmierzchowego. Ustaw za pomocą potencjometru P1 napięcie AIN1 = 2.5V. Potencjometrem P2 ustal odpowiednią czułość układu.

wy_komparatora **alias** P3.6

Do

```
If wy_komparatora = 0 Then 'włącz odbiornik
```

Else

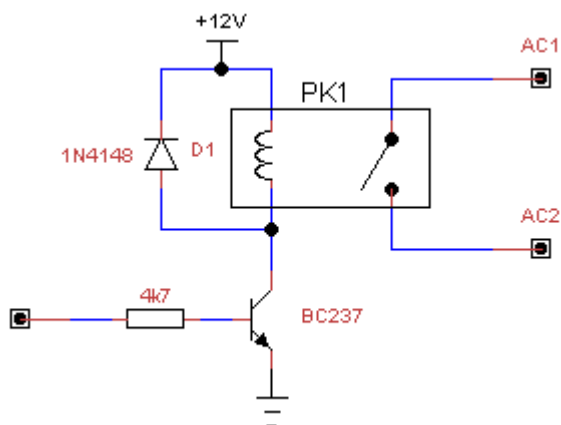
```
'wyłącz odbiornik
```

End if

Loop



Dołącz do linii P1.7 układ sterowania urządzeniami większej mocy z wykorzystaniem przekaźnika. Dorysuj obwód mocy wykorzystywany w ćwiczeniu.

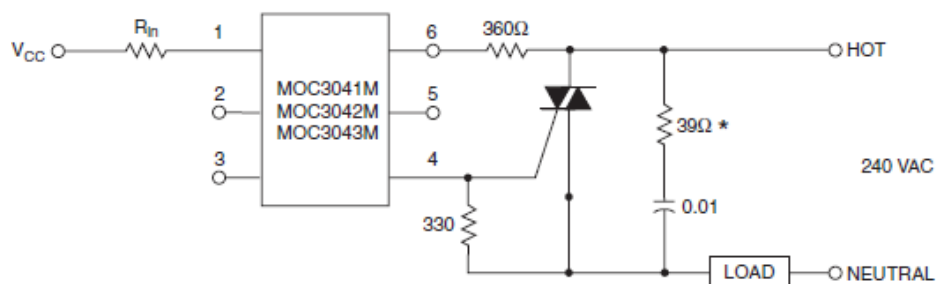


?

Do czego służy dioda D1?

i

W obwodach mocy stosuje się również sterowanie Triakiem. Optoizolację pomiędzy układem mikrokontrolera a napięciem sieciowym uzyskuje się za pomocą optotriaka np. MOC3043 przeznaczonego do pracy w obwodach 230 V, którego napięcie przebicia wynosi 7,5 kV. Ważną cechą tego optotriaka jest synchroniczne załączanie triaka przy przejściu napięcia sieci przez "0", co zapobiega powstawaniu szkodliwych zakłóceń w momencie załączania obwodu mocy. Wyjście mikrokontrolera podłączamy do wyprowadzenia 2 optotriaka (załączanie „0”). Rezystor R_{in} należy dobrać w taki sposób aby uzyskać odpowiedni prąd załączania (5mA – MOC3043, 10mA – MOC3043, 15mA – MOC3043)



6 Przetworniki A/C i C/A

Przetworniki analogowo-cyfrowe A/C (ang. *DAC – Digital to Analog Converter*) umożliwiają pomiar sygnałów (napięć) analogowych pochodzących np. z czujników i przetworników różnych wielkości fizycznych. Podstawowymi parametrami takich przetworników jest rozdzielczość czyli zdolność przetwornika do wiernego odtworzenia sygnału oraz szybkość jego działania. Większość obecnie produkowanych mikrokontrolerów posiada wbudowany jeden lub kilka przetworników A/C o rozdzielczości od 8, 10, 12 i więcej bitów. Mikrokontrolery rodziny AVR posiadają przeważnie jeden 10-bitowy przetwornik A/C, do którego można podłączać kilka źródeł sygnałów analogowych za pomocą sterowanego programowo multipleksera.

W ćwiczeniach wykorzystywany będzie mikrokontroler ATtiny26, którego przetwornik ADC posiada następujące cechy:

- rozdzielczość: 10-bitów
- precyzja: ± 2 LSB
- nieliniowość: 0.5 LSB
- czas konwersji: 13 - 260 μ s
- 11 multipleksowanych pojedynczych kanałów (ang. *Single Ended*)
- 8 kanałów różnicowych (ang. *Differential Input Channels*)
- 7 kanałów różnicowych z opcjonalnym wzmocnieniem x20
- zakres mierzonych napięć: 0 - AVCC
- wybieralne źródło napięcia odniesienia (wewnętrzne 2,56V, VCC i zewnętrzne)
- tryby pracy: ciągły (ang. *Free Running*) i pojedyncza konwersja (ang. *Single Conversion*)
- obsługa za pomocą przerwań
- tryb uśpienia i eliminacji szumu

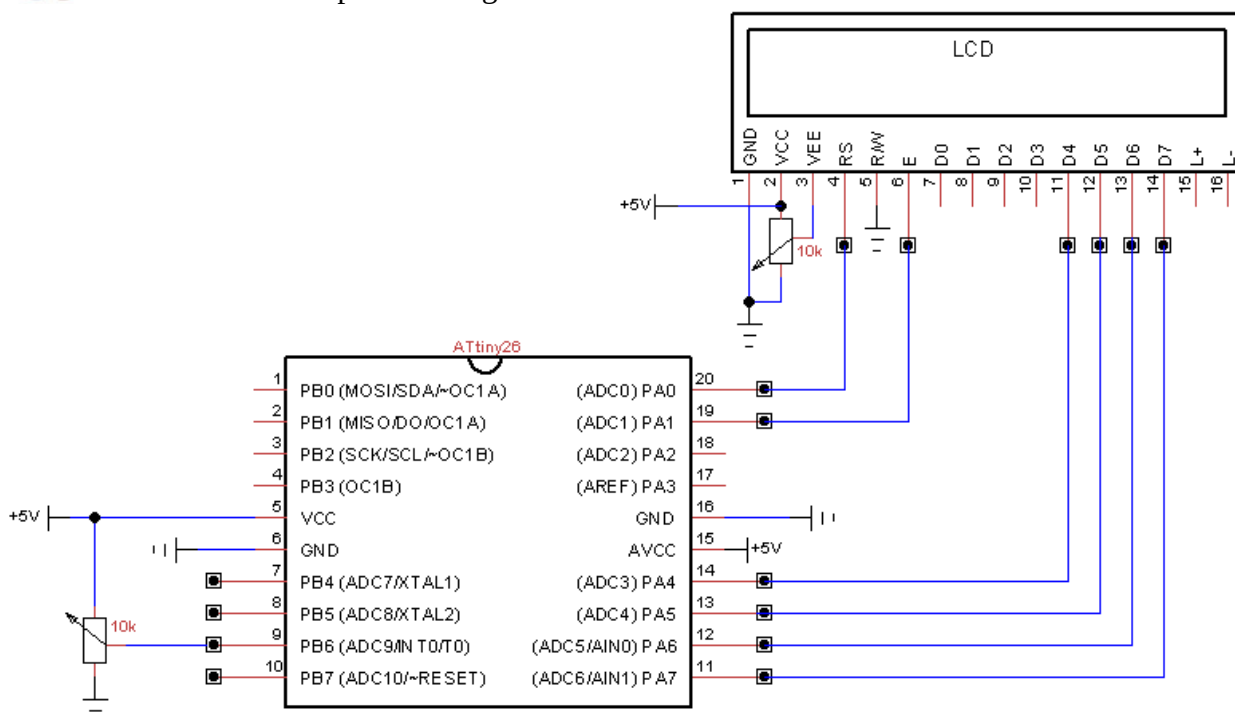
6.1 Przetwornik A/C – woltomierz cyfrowy

Data:

Ocena:



Schemat układu pomiarowego



i

Wynik konwersji 10 bitowego przetwornika ADC znajduje się w dwóch rejestrach (ADCH i ADCL).

Przy pomiarze pojedynczego wejścia (Single Ended) zawartość tych rejestrów można wyznaczyć ze wzoru:

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

Dla wejść różnicowych równanie przyjmuje postać:

$$ADC = \frac{(V_{IN+} - V_{IN-}) \cdot 1024 \cdot GAIN}{V_{REF}}$$



Oblicz wynik konwersji (zawartość ADC) dla poniższych danych.

$V_{IN} = 1V$,
 $V_{REF} = 2,56V$

ADC =



Jakie napięcie V_{IN} podłączono do wejścia przetwornika A/C jeśli wynik konwersji wyniósł ADC=512, przy napięciu referencyjnym 5V?

.....

.....

.....



Uruchom program testujący działanie przetwornika A/C mikrokontrolera ATtiny26.

```

Config Porta = Output
'konfiguracja ADC: konwersja pojedyncza, VREF=2,56V
Config Adc = Single , Prescaler = Auto , Reference = Internal
Start Adc

Dim U As Integer, X as byte

Cls
Cursor Off Noblink
Locate 1 , 1
Lcd "U=0.00V"

Do
    U = Getadc(9)      'wykonanie konwersji, kanał=9
    U = U / 4
    Locate 1 , 6
    X = U Mod 10      'najmniej znacząca cyfra
    Lcd X
    U = U/10
    X = U Mod 10      'bardziej znacząca cyfra
    Locate 1 , 5
    Lcd X
    U = U/10          'najbardziej znacząca cyfra
    Locate 1 , 3
    Lcd X
Loop

```



.....

.....

.....

.....

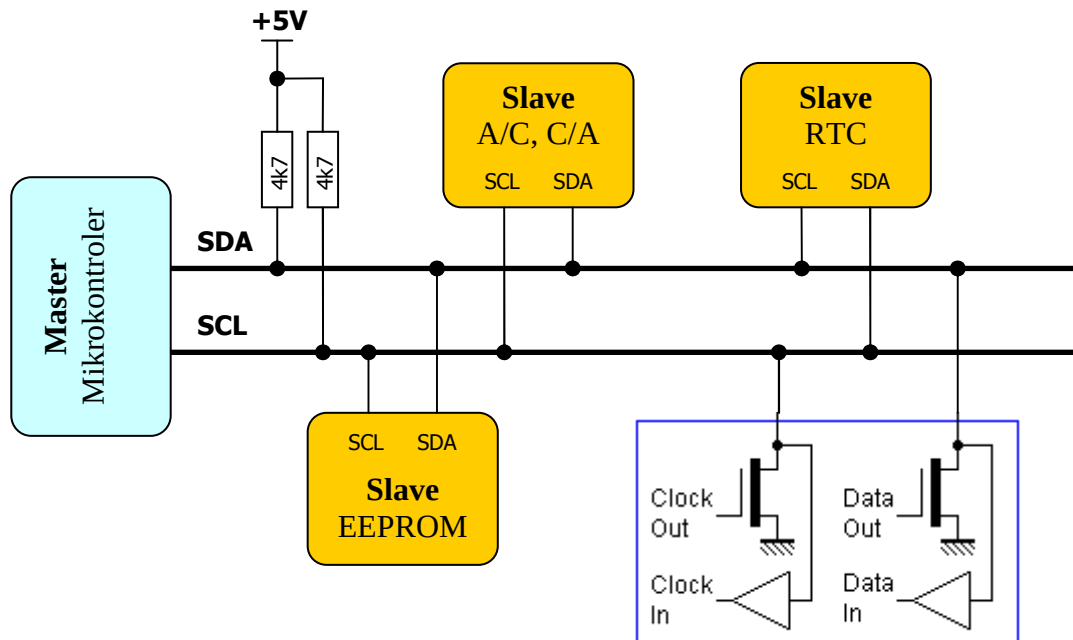
.....

7 I²C

I²C – szeregowa, dwukierunkowa magistrala służąca do przesyłania danych w urządzeniach elektronicznych. Została opracowana przez firmę Philips na początku lat 80. Znana również pod akronimem IIC.

Cechy charakterystyczne szyny I2C:

- transmisja odbywa się za pomocą dwóch linii sygnałowych: SDA - linii danych i SCL - linii sygnału taktującego, niewielka liczba linii upraszcza układ połączeń i zmniejsza liczbę potrzebnych linii portów mikrokontrolera,
- każdy z układów dołączonych do szyny posiada niepowtarzalny określony sprzętowo adres, umożliwiając jednoznaczne zaadresowanie danego urządzenia do nadawania lub odbioru,
- przesyłaniem danych steruje układ nadrzędny MASTER, przy czym w systemie może występować więcej niż jeden układ nadrzędny,
- w przypadku jednoczesnego rozpoczęcia nadawania przez kilka urządzeń nadrzędnych, mechanizm arbitrażu nie dopuszcza do utraty, ani przekłamania przesyłanych danych,
- synchronizacja transmisji za pomocą sygnału taktującego umożliwia komunikację między urządzeniami stosującymi różne prędkości przesyłania danych,
- układy powinny posiadać wyjścia typu otwarty kolektor/dren,
- maksymalna prędkość transmisji danych po szynie I2C wynosi 100 kb/s w standardowym trybie pracy szyny i 400 kb/s w trybie szybkim,
- liczba układów podłączonych do szyny ograniczona jest trybem adresowania i maksymalną pojemnością szyny równą 400pF.



7.1 PCF8591- karta przetworników A/C i C/A

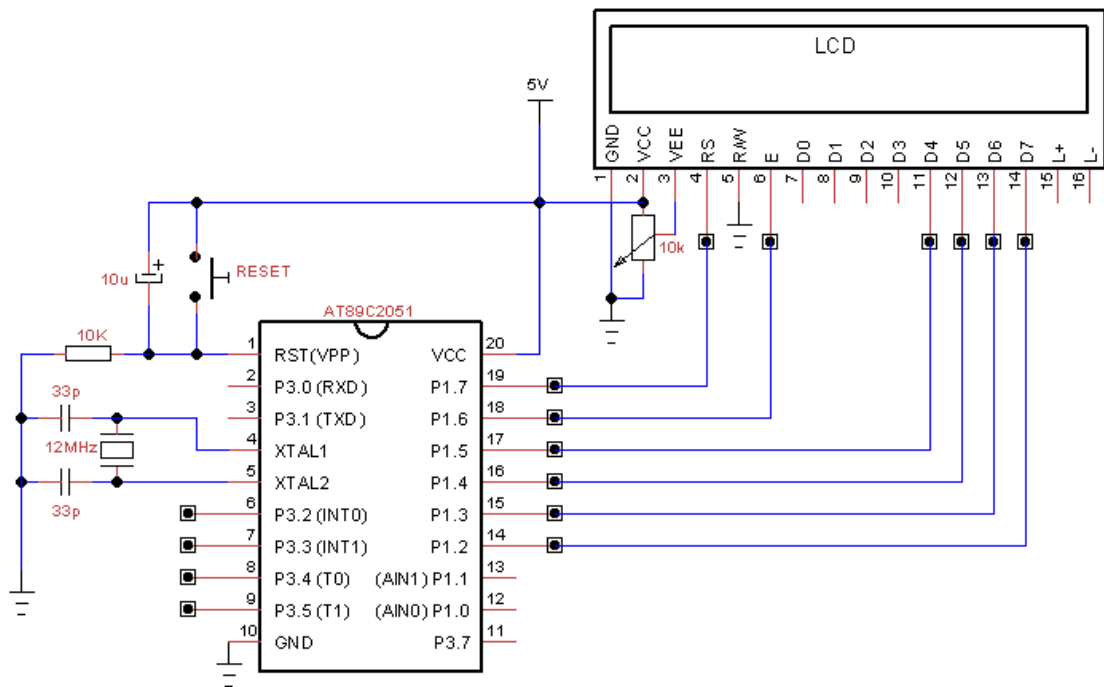
Data:

Ocena:

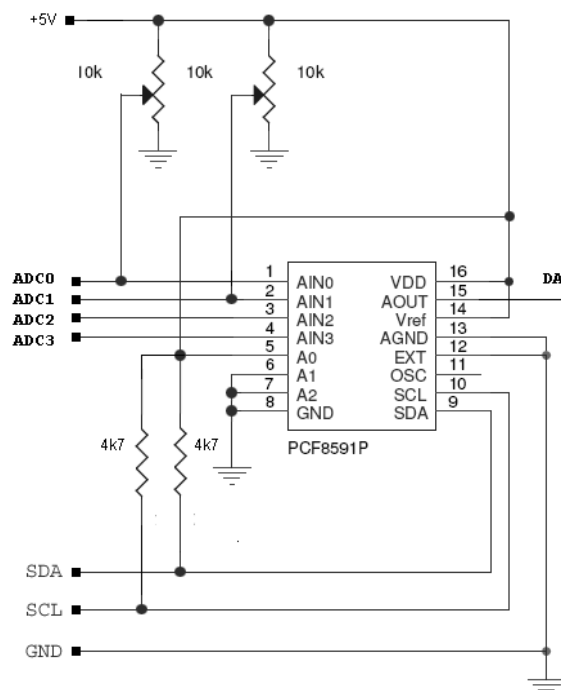


Schemat układu pomiarowego

a) układ obsługi magistrali I2C i prezentacji wyniku



b) karta przetworników A/C i C/A PCF8591 (linie SDA i SCL powinny być podciągnięte do +5V rezystorami 4k7)





Zapoznaj się z dokumentacją układu PCF8591. Uruchom program testowy karty przetworników. Dodaj stosowne komentarze.

```

Declare Sub Pcf8591
Config Sda = P3.4
Config Scl = P3.5
Config I2cdelay = 100
I2cinit

Dim U           As Byte
Dim U_0         As Byte
Dim U_1         As Byte
Dim Kanal       As Byte
Dim Dac         As Byte

Dim Pcf8591w    As Const &B10010000    'PCF8591 adres zapisu.
Dim Pcf8591r    As Const &B10010001    'PCF8591 adres odczytu.
Dim Kanal_0     As Const &B01000000    'DAC + Kanal 0
Dim Kanal_1     As Const &B01000001    'DAC + Kanal 1

dac = 0
Do
    Kanal = Kanal_0
    Call Pcf8591
    U_0 = U
    Kanal = Kanal_1
    Call Pcf8591
    U_1 = U
    Cls
    Home
    LCD "U0="; U_0; "U1="; U_1
    Waitms 100
Loop
End

Sub Pcf8591
    I2cstart
    I2cwbyte Pcf8591w    'adres zapisu.
    I2cwbyte Kanal      'wybór kanału
    I2cwbyte Dac        'ustalenie przetwornika DAC

    I2cstart
    I2cwbyte Pcf8591r    'odczyt Kanału ADC
    I2crbyte U , Ack
    I2crbyte U , Nack
    I2cstop
End Sub

```



.....

.....

.....

.....

.....

7.2 AT24C04 – pamięć EEPROM

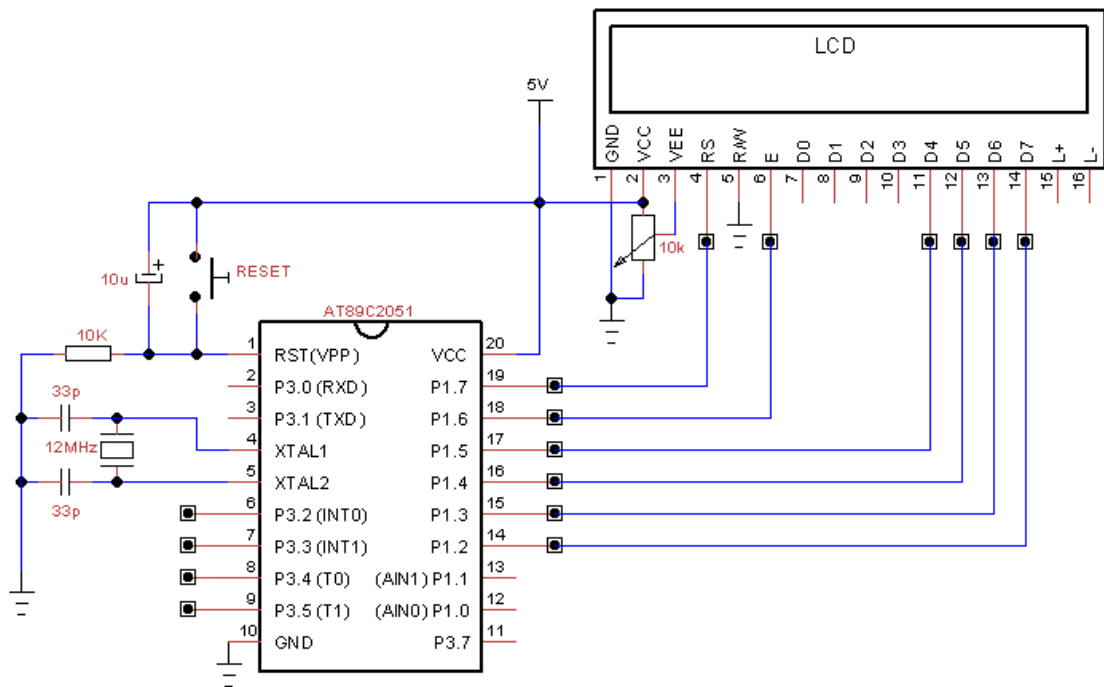
Data:

Ocena:

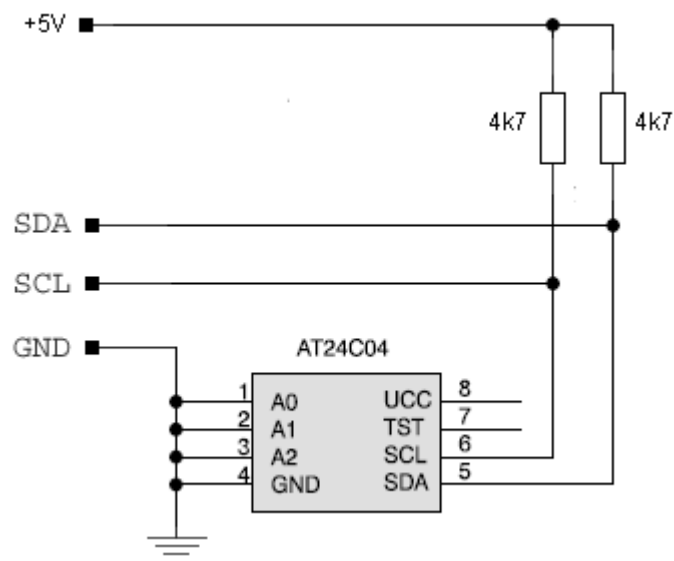


Schemat układu pomiarowego

a) układ obsługi magistrali I2C i prezentacji wyniku



b) karta pamięci EEPROM AT24C04 (linie SDA i SCL powinny być podciągnięte do +5V rezystorami 4k7)





Mikrokontrolery AVR (ATtiny, ATmega) posiadają moduł wewnętrznej pamięci EEPROM o pojemnościach od 128B (ATtiny26) do 4KB (Atmega128).



Zapoznaj się z dokumentacją układu AT24C04. Uruchom program testowy karty pamięci EEPROM. Dodaj stosowne komentarze.

```
'konfiguracja magistrali I2C
Config Sda = P3.5
Config Scl = P3.7

'deklaracje procedur i zmiennych
Declare Sub Rd_eeeprom(adres As Byte, Dana As Byte)
Declare Sub Wr_eeeprom(adres As Byte, Dana As Byte)
Dim Adres As Byte, Dana As Byte

Adres = 0
'skomentuj poniższe dwie linie, uruchom mikrokontroler ponownie
Dana = 123
Call Wr_eeeprom Adres, Dana
Do
    Call Rd_eeeprom Adres, Dana
    Cls
    Home
    LCD "EEPROM["; Adres; "]="; Dana
    Wait 1
Loop

'procedury odczytu i zapisu EEPROM 24C04
Sub Rd_eeeprom(adres As Byte, Dana As Byte)
    I2Cstart
    I2Cwbyte 160
    I2Cwbyte Adres
    I2Cstart
    I2Cwbyte 161
    I2Crbyte Dana, 9
    I2Cstop
End Sub

Sub Wr_eeeprom(adres As Byte, Dana As Byte)
    I2Cstart
    I2Cwbyte 160
    I2Cwbyte Adres
    I2Cwbyte Dana
    I2Cstop
    Waitms 10
End Sub
```



.....

.....

.....

.....

.....

8 1-Wire

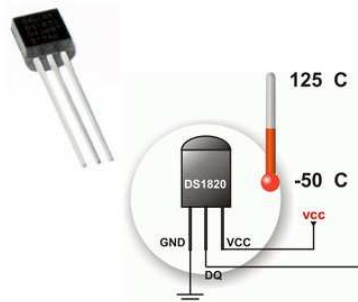
Interfejs 1-Wire opracowany został przez firmę Dallas. Jego nazwa wywodzi się z faktu, że do komunikacji używana jest tylko jedna linia danych (oraz linia zerowa). Jest on używany do komunikacji pomiędzy niewielkimi urządzeniami, takimi jak: termometry cyfrowe, instrumenty metrologiczne, zamki elektroniczne typu iButton, itp. Magistrala 1-Wire jest przeznaczona do przesyłania informacji pomiędzy układem nadrzędnym master - mikrosterownik a układami podrzędnymi slave (np. termometry, układy identyfikacji, pamięci SRAM i EEPROM, programowalne klucze). Transmisja odbywa się w obu kierunkach z wykorzystaniem jednego przewodu sygnałowego (oprócz masy), który jednocześnie może być wykorzystany do zasilania układów do niego podłączonych. Dane przesyłane są z prędkością od bliskiej 0 do 16,3 kbps w trybie standard oraz do 115 kbps w trybie overdrive. Każde z urządzeń podłączonych do magistrali musi mieć wyjście typu otwarty dren, a linia sygnałowa DQ jest połączona do zasilania przez rezystor podciągający o wartości około 5k Ω . Zatem w stanie bezczynności linia DQ jest w stanie wysokim. Magistrala nie ma ustalonego formatu danych. Sposób przesyłania informacji zależy od konfiguracji układów podrzędnych. Zawsze jednak jako pierwszy jest przesyłany bit najmniej znaczący.



Architektura magistrali 1-Wire

Przykładowe układy 1-Wire

- **DS1820** – Czujnik temperatury



- **DS1990A** – pastylka identyfikacyjna iButton



8.1 DS1820 - pomiar temperatury

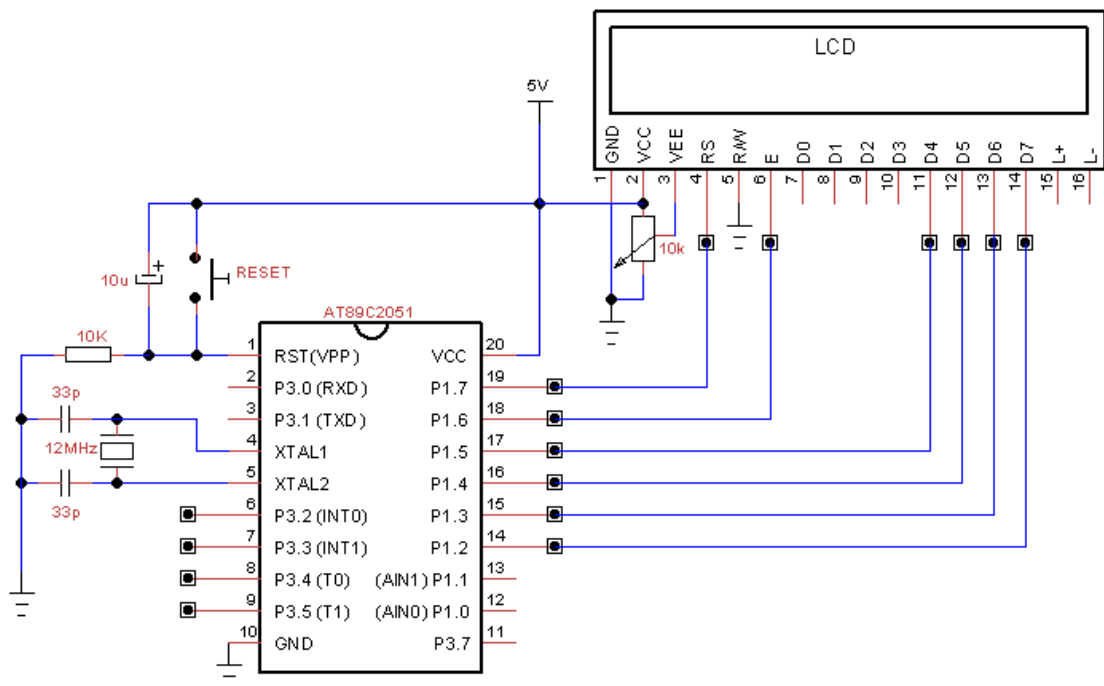
Data:

Ocena:

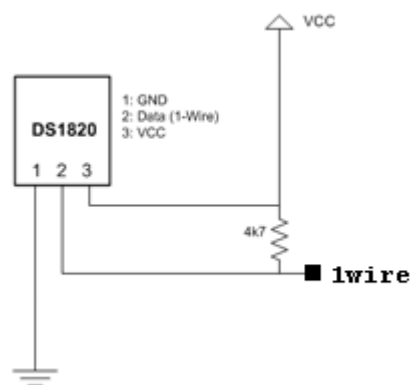


Schemat układu pomiarowego

a) układ obsługi magistrali 1-Wire i prezentacji wyniku



a) układ pomiaru temperatury na układzie DS1820





Zapoznaj się z dokumentacją układu DS1820. Uruchom program pomiaru temperatury. Dodaj stosowne komentarze.

```
Dim Bd(9) As Byte
Dim T As Integer
Declare Sub Read1820
Config 1wire = P3.4

Do
    Read1820
    Cls
    LCD "T="; T
Loop

Sub Read1820
    1wreset
    1wwrite &HCC
    1wwrite &H44
    Wait 1
    1wreset
    1wwrite &HCC
    1wwrite &HBE
    Bd(1) = 1wread(9)
    T = Makeint(bd(1) , Bd(2))
    T = T * 10
    T = T / 16
End Sub
```



Jaka linia mikroprocesora obsługuje magistralę 1Wire? Do czego służy opóźnienie Wait 1?

.....

.....

.....



.....

.....

.....

.....

.....

9 DODATKI

9.1 KODY ASCII

Tablica podstawowych kodów ASCII:

*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Przykład: kod 30H oznacza znak "0"

Rozszerzone kody ASCII:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ç	ü	é	â	ä	à	ç	ê	ë	è	ï	î	ì	ñ		
9	É	æ	Æ	ô	ö	ò	û	ÿ	õ	ü	ç	£	¥	₹	ƒ	
A	á	í	ó	ú	ñ	Ñ	æ	œ	¿	ı	½	¾	ı	«	»	
B	⌘	⌘	⌘													
C	⌘	⌘	⌘													
D	⌘	⌘	⌘													
E	α	β	Γ	Π	Σ	σ	μ	τ	ϑ	θ	Ω	δ	ω	ϖ	€	π
F	≡	±	≥	≤	ƒ	J	÷	≈	°	·	·	√	°	²	³	⁴