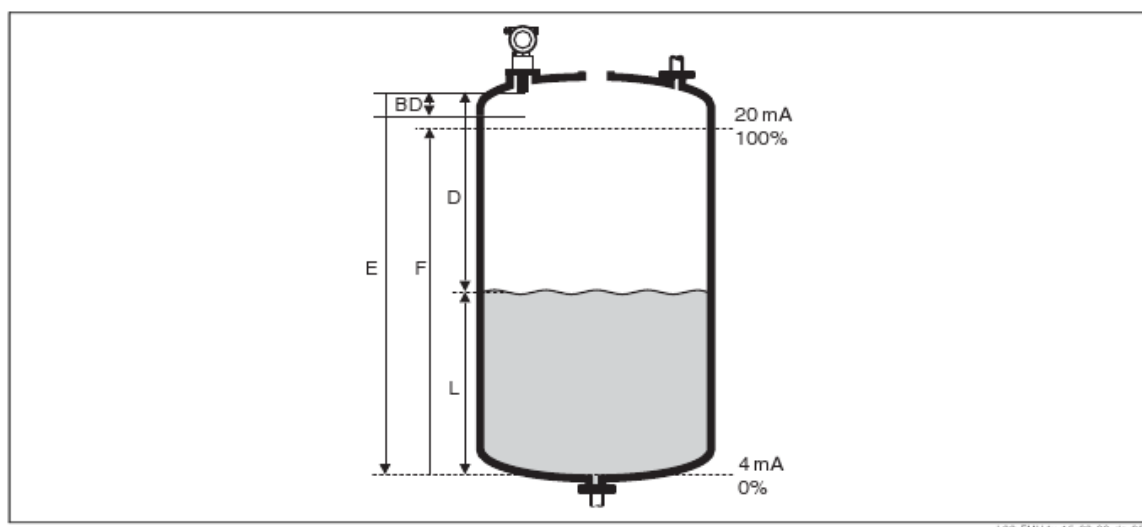


Rozdział 1

Pomiar poziomu cieczy



E: Wysokość zbiornika; F: Pełny zakres pomiaru; D: Odległość od membrany czujnika do poziomu produktu; L: Poziom produktu; BD: Strefa martwa

Rysunek 1.1 Pomiar poziomu cieczy [1]

Pomiar poziomu (zarówno ciecży jak i materiałów sypkich) najczęściej sprowadza się do pomiaru długości lub odległości. W zależności od mierzonego medium zakresy pomiarowe mogą wynosić od kilku centymetrów (np. w samochodowych zbiornikach paliwa) do kilkudziesięciu metrów (np. w silosach zbożowych). W przemyśle najczęściej dokonuje się pomiarów poziomu ciecży, past, szlamów, gazów ciekłych, drobnoziarnistych materiałów sypkich i kruszyw, zgromadzonych w zbiornikach stacjonarnych i przenośnych, magazynowych i buforowych. Pomiarów tego typu wykorzystywane są praktycznie w każdej gałęzi przemysłu, dlatego też środowisko w jakim są one wykonywane jest bardzo zróżnicowane. Pomiarów dokonuje się przy temperaturach zarówno bardzo niskich (bliskich zeru bezwzględnemu) jak i wysokich (kilkaset stopni Celsjusza). Ciśnienie panujące w zbiornikach jest również bardzo zróżnicowane.

Istnieje wiele różnych metod pomiaru, które są uzależnione najczęściej od charakteru mierzonego medium. Najczęściej stosowanymi metodami są ultradźwiękowa, elektromechaniczna, pojemnościowa, przewodnościowa, różnicy ciśnień, hydrostatyczna radiometryczna, optyczna i inne.

1.1 Metoda ultradźwiękowa



Rysunek 1.2 Pomiar ultradźwiękowy [1]

Jest to jedna z częściej spotykanych metod pomiaru. Nadajnik czujnika emituje krótkie impulsy ultradźwiękowe, które po odbiciu od powierzchni produktu wracają do anteny, pracującej jednocześnie jako odbiornik. Zasada działania urządzenia opiera się na pomiarze czasu przelotu t fali akustycznej pomiędzy czujnikiem a powierzchnią medium. Czujnik wykorzystuje zmierzony czas t oraz informację o prędkości dźwięku c do wyliczenia odległości D zgodnie z zależnością:

$$D = c \cdot \frac{t}{2} \quad (1.1)$$

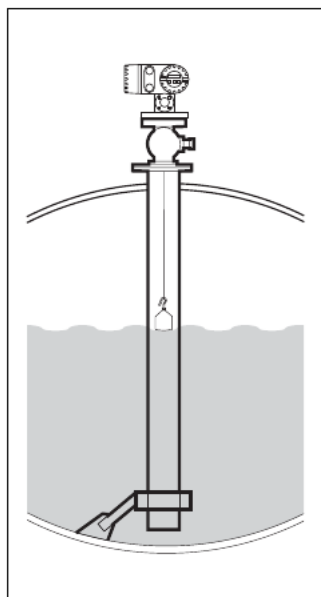
gdzie c - prędkość dźwięku

Informacja o wysokości zbiornika E pozwala na wyliczenie poziomu L z równania 1.2:

$$L = E - D \quad (1.2)$$

Czujniki tego typu często posiadają wbudowany czujnik temperatury służący do kompensacji zmian prędkości rozchodzenia się fali ultradźwiękowej przy zmianach temperatury.

1.2 Metoda elektromechaniczna

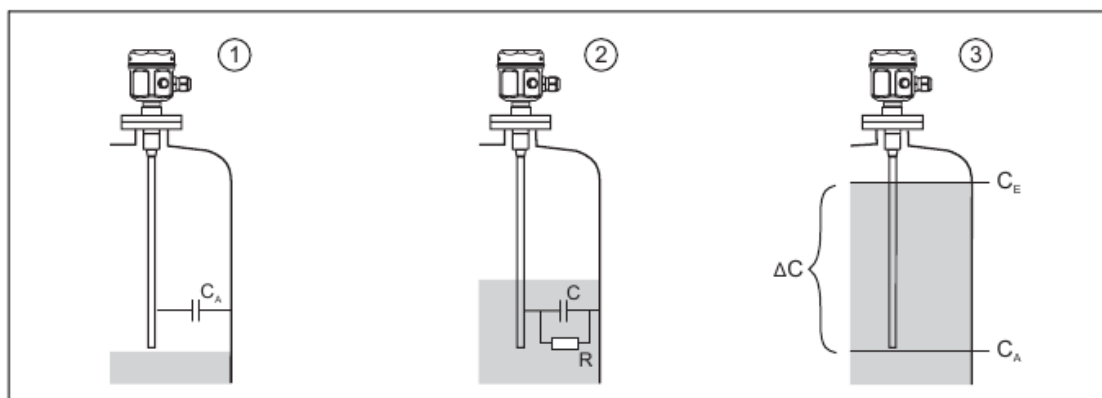


Rysunek 1.3 Pomiar elektromechaniczny [1]

Zasada działania pływakowego przetwornika poziomu oparta jest na pomiarze przesunięcia czujnika wypieranego przez ciecz. Mały czujnik pływakowy jest dokładnie pozycjonowany w cieczy za pomocą serwomechanizmu elektromagnetycznego. Jest on zawieszony na linie pomiarowej, nawiniętej na precyzyjnie rowkowany bęben, umieszczony w obudowie. Bęben jest sterowany przez serwomechanizm ze sprzężeniem magnetycznym, całkowicie odizolowany od obudowy bębna. Magnesy zewnętrzne znajdują się na bębnie, natomiast magnesy wewnętrzne - na elektrycznym silniku napędowym. Obrót magnesów wewnętrznych, na skutek przyciągania magnetycznego, powoduje również obracanie się magnesów zewnętrznych, a tym samym ruch obrotowy całego zespołu bębna. Masa pływaka zawieszona na linie wytwarza moment obrotowy oddziaływający na magnesy zewnętrzne, powodując tym samym zmianę strumienia magnetycznego. Zmiany strumienia, generowane pomiędzy elementami zespołu bębna, wykrywane są przez przetwornik elektromagnetyczny o unikatowej konstrukcji, znajdujący się na magnecie wewnętrznym. Utrzymanie sygnału sterującego, generowanego przez zmiany strumienia magnetycznego, na wymaganym poziomie, tj. zgodnym z wprowadzoną nastawą jest zapewnione przez właściwe kontrolowanie silnika napędowego.

W czasie opuszczania czujnika pływakowego, po zetknięciu z cieczą jego masa ulega zmniejszeniu ze względu na siłę wyporu cieczy. W efekcie, moment obrotowy mechanizmu ze sprzężeniem magnetycznym ulega zmianie, która mierzona jest przez 5 par czujników Halla z kompensacją temperaturową. Sygnał odzwierciedlający pozycję pływaka jest przesyłany do układu sterowania silnikiem. Podczas, gdy poziom cieczy podnosi się i opada, pozycja pływaka jest regulowana linką przez silnik napędowy. Poziom cieczy jest wyznaczany z dokładnością ok 1mm , na podstawie dokładnej analizy ruchu obrotowego bębna z nawiniętą linką pomiarową.

1.3 Metoda pojemnościowa



R : Przewodność cieczy

C : Pojemność cieczy

C_A : Pojemność początkowa (sonda odkryta)

C_E : Pojemność końcowa (sonda zakryta): Zmiana pojemności

ΔC : Zmiana pojemności

Rysunek 1.4 Pomiar pojemnościowy [1]

Zasada pomiaru metodą pojemnościową bazuje na wyznaczaniu zmiany pojemności kondensatora, którego jedną elektrodę stanowi sonda, natomiast drugą ściana zbiornika lub rura osłonowa (materiał przewodzący). Zmiana poziomu cieczy powoduje zmianę pojemności utworzonego w ten sposób kondensatora. Podczas, gdy sonda znajduje się w powietrzu, mierzona jest określona pojemność początkowa. Po napełnieniu zbiornika cieczą, pojemność kondensatora wzrasta proporcjonalnie do stopnia zakrycia sondy.

Dla cieczy o przewodności większej od $100 \frac{\mu S}{cm}$, pomiar jest niezależny od wartości stałej dielektrycznej (DK) medium. W efekcie, zmiany wartości DK nie wpływają na wskazanie wartości mierzonej. Ponadto, w przypadku sond z częścią nieaktywną, konstrukcja przyrządu pozwala wyeliminować wpływ osadu i kondensacji przy przyłączu technologicznym.

1.4 Metoda przewodnościowa

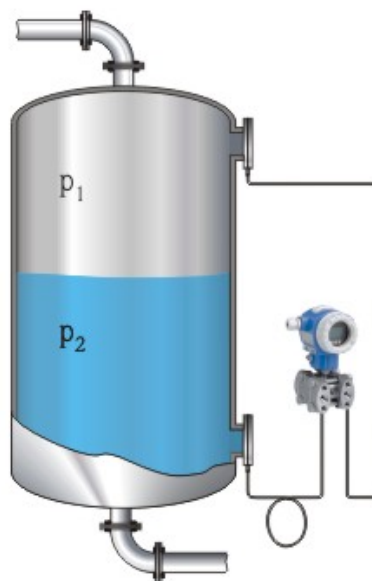


Rysunek 1.5 Pomiar przewodnościowy [1]

W pustym zbiorniku między czujnikami prętowymi występuje różnica potencjałów. Po napełnieniu zbiornika, między elektrodą odniesienia, i na przykład, czujnikiem prętowym maksymalnego poziomu, popłynie prąd i nastąpi przełączenie wyjścia sygnalizatora. Natychmiast po odkryciu przez ciecz tego czujnika następuje przełączenie do stanu początkowego. W przypadku regulacji dwustanowej, przełączenie urządzenia do stanu początkowego następuje dopiero po odsłonięciu obu czujników - stanu maksymalnego i minimalnego.

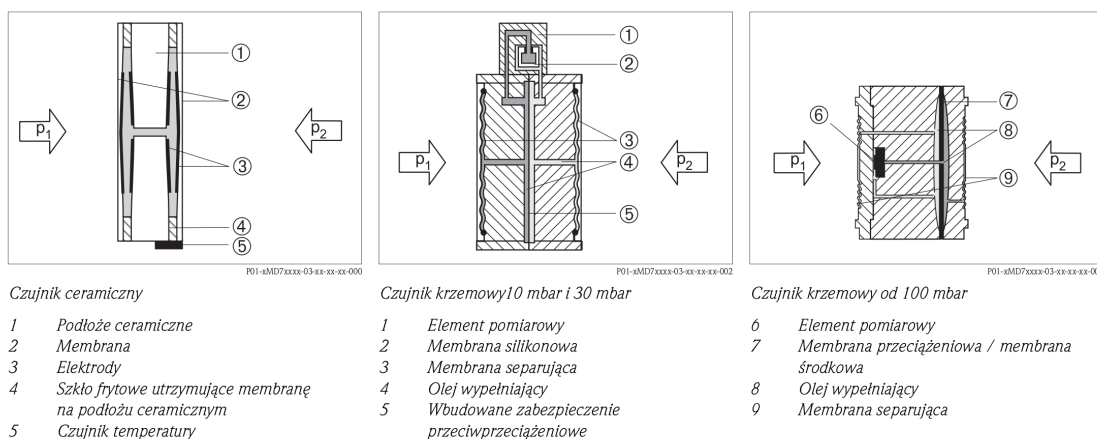
Stosowanie napięcia przemiennego zapobiega korozji czujników prętowych lub ich zniszczeniu na skutek elektrolizy. Materiał, z którego wykonane są ścianki zbiornika nie jest istotny, ponieważ system został zaprojektowany w postaci zamkniętego obwodu bezpotencjałowego obejmującego czujniki prętowe i moduł elektroniki. Nie ma niebezpieczeństwa, jeśli w czasie pracy czujniki prętowe zostaną dotknięte.

1.5 Metoda różnicy ciśnień



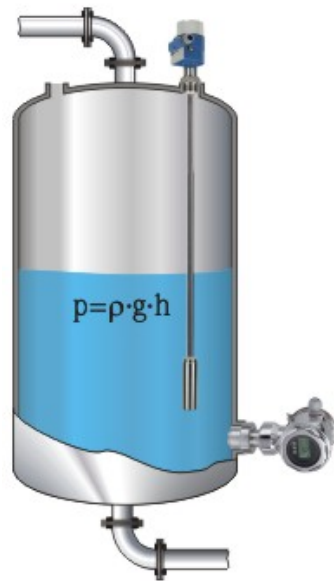
Rysunek 1.6 Pomiar różnicy ciśnień [1]

Ciśnienie procesowe działające na membranę czujnika, powoduje jej ugięcie. Zmiana odległości między bardzo precyzyjnie naniesionymi złotymi elektrodami, powoduje zmianę pojemności po obydwu stronach. Na tej podstawie wyznaczany jest stosunek pomiędzy nadciśnieniem a ciśnieniem hydrostatycznym (aby pomiar był dokładny, nie powinien być większy niż 4 : 1). Metoda ta ma przewagę nad metodą hydrostatyczną gdyż na wynik pomiaru nie wpływa nadciśnienie w zbiorniku. Jest to główny powód, dla którego metodę tę stosuje się w zbiornikach ciśnieniowych.



Rysunek 1.7 Spotykane konstrukcje przetworników ciśnień [1]

1.6 Metoda hydrostatyczna



Rysunek 1.8 Pomiar hydrostatyczny [1]

Ciężar słupa cieczy powoduje powstanie ciśnienia hydrostatycznego działającego na membranę czujnika. Zakładając stałą gęstość produktu ciśnienie hydrostatyczne jest liniową funkcją wysokości słupa cieczy:

$$p_{hydrostat} = \rho \cdot g \cdot h \quad (1.3)$$

ρ = gęstość

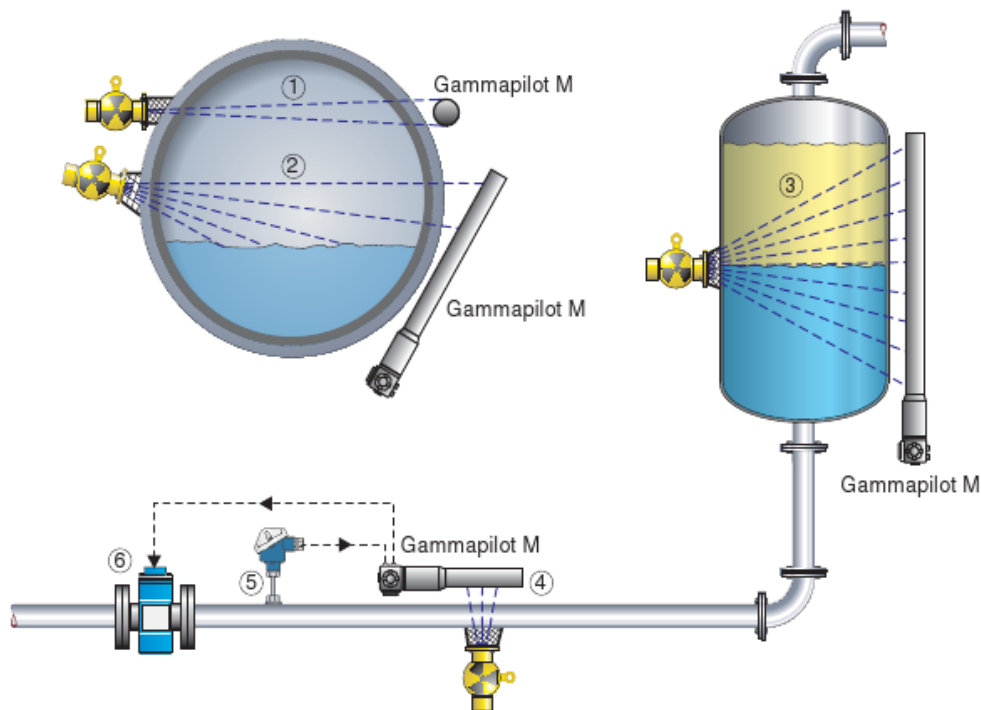
g = stała grawitacji ($9.81 \frac{m}{s^2}$)

h = odległość pomiędzy powierzchnią cieczy, a środkiem membrany czujnika

Zaletami tej metody pomiaru są:

- prosta konstrukcja układu pomiarowego
- wewnętrzne elementy zbiornika, zapienienie i turbulencje powierzchni cieczy nie zakłócają pomiaru

1.7 Metoda radiometryczna



Rysunek 1.9 Pomiar radiometryczny [1]

Ta metoda pomiaru stosowana jest jedynie w ekstremalnych warunkach procesowych bądź tam, gdzie zastosowanie innych metod pomiaru jest niemożliwe.

Zasada pomiaru tą metodą polega na wykrywaniu absorpcji promieniowania γ przechodzącego przez mierzone medium. Źródło promieniowania jonizującego (w postaci izotopu cezu lub kobaltu) umieszcza się na ścianie zbiornika, fale przechodząc przez medium ulegają osłabieniu na skutek absorpcji. Zamontowany po przeciwległej stronie zbiornika detektor wykrywa wiązkę promieniowania wykorzystując efekt fotoluminescencji. Przykładowe techniki pomiaru tą metodą z wykorzystaniem czujnika Gammapiilot M firmy Endress+Hauser znajdują się na rysunku 1.9.

Jak podaje producent, czujniki tego typu można również wykorzystać do następujących celów¹:

- pomiar poziomu, również w trybie kaskadowym (konfiguracja szeregową przetworników) lub zwielokrotniania czułości (konfiguracja równoległa przetworników)
- sygnalizację poziomu (monitorowanie poziomu minimalnego lub maksymalnego)
- detekcja rozdziału faz
- pomiar gęstości
- pomiar koncentracji
- pomiar przepływu masowego z wykorzystaniem przepływomierza objętościowego

¹Więcej informacji na stronie producenta: http://www.pl.endress.com/gammapiilot_m

Bibliografia

- [1] Endress+Hauser - aparatura pomiarowa. <http://www.pl.endress.com/>.
- [2] Strona domowa programu AVRStudio. <http://www.atmel.com/avrstudio/>.
- [3] Wikipedia, wolna encyklopedia. <http://pl.wikipedia.org/>.
- [4] Atmel. *AVR ATmega16 Datasheet (Dokumentacja techniczna procesora Atmel ATmega16)*. Atmel Corporation, wydanie 2466p-avr-08/07, 2007.
- [5] R. Baranowski. *Mikrokontrolery AVR ATmega w praktyce*. Wydawnictwo BTC, Warszawa, wydanie 1, 2005.
- [6] J. Brzózka. *Regulatory i układy automatyki*. Mikom, Warszawa, wydanie 1, 2004.
- [7] J. Halawa. *Wyznaczanie parametrów regulatorów na podstawie transmitancji układu zamkniętego*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, wydanie 1, 2004.
- [8] J. Halawa. *Symulacja i komputerowe projektowanie dynamiki układów sterowania*. Oficyna Wydawnicza Politechniki Wrocławskiej, wydanie 1, 2007.
- [9] J. Klimesz, W. Solnik. *Urządzenia automatyki*. Wydawnictwo Politechniki Wrocławskiej, Wrocław, wydanie 1, 1991.
- [10] A. Królikowski, D. Horla. *Identyfikacja obiektów sterowania : metody dyskretne*. Wydawnictwo Politechniki Poznańskiej, Poznań, wydanie 1, 2005.
- [11] B. Łysakowska, G. Mzyk. *Komputerowa symulacja układów automatycznej regulacji w środowisku MATLAB/Simulink*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, wydanie 1, 2005.
- [12] K. Mańczak, Z. Nahorski. *Komputerowa identyfikacja obiektów dynamicznych*. Państwowe Wydawnictwo Naukowe, Warszawa, 1983.
- [13] P. Prof. Zoran Vikić. *Lectures on PID controllers*. University of Zagreb, 2002. http://arri.uta.edu/acs/jyotirmay/EE4343/Labs_Projects/pidcontrollers.p%df.
- [14] Siemens. *SIMATIC S7-200 Programmable Controller System Manual*. Siemens AG, 2005.
- [15] W. yong Han, J. wook Han, C. goo Lee. *Development of a Self-tuning PID Controller based on Neural Network for Nonlinear Systems*. Dept.of Electrical Engineering, Jeonju Technical College, 1999. <http://med.ee.nd.edu/MED7/med99/papers/MED182.pdf>.

-
- [16] Z. Zajda, L. Żebrowski. *Urządzenia i układy automatyki*. Wydawnictwo Politechniki Wrocławskiej, Wrocław, 1993.

Dodatek A

Kod programu przetwornika

```
1 #include <inttypes.h>
2 #include <avr/io.h>
3 #include <avr/interrupt.h>
4 #include <util/delay.h>
5 #include "dm_lcd.h"
6
7 #define PWM_out(value)      OCR1A=value
8
9
10 volatile uint16_t freq;           // zmienna okreslajaca
    czestotliwosc
11 volatile uint16_t adcvalue;      // przerworzona wartosc z
    ADC
12 uint16_t params[10];            // tablica parametrow
    programu
13
14 SIGNAL(SIG_ADC) {               // przerwanie z
    przetwornika ADC
15     adcvalue = ADCW;            // czytaj wartosc z
    przetwornika ADC
16 }
17
18 void lcd_puti(uint8_t value) {    // wyswietla liczbe 8
    bitowa na LCD
19     char string[3];
20     if ( value < 10 ) { lcd_puts(" "); }
21     if ( value < 100 ) { lcd_puts(" "); }
22     itoa(value, string, 10);
23     lcd_puts(string);
24 }
25
26 void lcd_putd(uint16_t value) {   // wyswietla liczbe 16
    bitowa na LCD
27     char string[5];
28     if ( value < 10 ) { lcd_puts(" "); }
29     if ( value < 100 ) { lcd_puts(" "); }
30     if ( value < 1000 ) { lcd_puts(" "); }
31     //if ( value < 10000 ) { lcd_puts(" "); } // i tak wykorzystujemy
    liczby max 10b
32     itoa(value, string, 10);
33     lcd_puts(string);
34 }
35
```

```

36 int kbb_read() { // obsluga klawiatury
37
38     uint8_t read; // inicjalizacja zmiennej
        tymczasowej
39     read = ( PINC & DDRC ); // odczyt portu C,
        maskowanie rejesterm kirunkowym
40
41     if (read == 4) { return 1; } // zwracanie odczytu
42     if (read == 8) { return 2; }
43     if (read == 16) { return 3; }
44     if (read == 32) { return 4; }
45
46     return 0;
47 }
48
49 void pwm_init(void) { // inicjalizacja PWM
50     // Ustaw licznik 1 w trybie 10 bitowego wyjscia PWM
51     //TCCR1A = _BV(COM1A1)|_BV(COM1B1)|_BV(WGM10)|_BV(WGM11); // 10
        bit pwm
52     TCCR1A = _BV(COM1A1)|_BV(COM1B1)|_BV(WGM11); // 10bit fast pwm
53
54     // Ustaw czestotliwosc licznika zgodna z czestotliwoscia taktownia
        zegara
55     //TCCR1B = _BV(CS10);
56     TCCR1B = _BV(CS10)|_BV(WGM12)|_BV(WGM13);
57
58     ICR1 = 1; // prescaler
59 }
60
61 void adc_init(void) { // inicjalizacja ADC
62     // wlacz przetwornik adc i uruchom generowanie przerwan
63     // czestotilwosc taktowania F_ADC=F_CPU/64(125 kHz)
64     ADCSRA = _BV(ADEN)|_BV(ADIF)|_BV(ADPS2)|_BV(ADPS1);
65
66     // Bit | Nazwa | Opis
67     // 0 | ADPS0 | preskaler
68     // 1 | ADPS1 | preskaler
69     // 2 | ADPS2 | preskaler
70     // 3 | ADIF | generowanie przerwania przez przetwornik
71     // 4 | ADIF | znacznik przerwania z przetwornika
72     // 5 | ADFR | przetwarzanie samodzielne (Free Run)
73     // 6 | ADSC | start konwersji – jesli jest ustawiony ten bit oraz
        ADEN
74     // 7 | ADEN | wlaczenie przetwornika (AD Enable)
75     //
76     // Preskaler:
77     // ADPS2 ADPS1 ADPS0 podzial
78     // -----
79     // 0 0 0 1
80     // 0 0 1 2
81     // 0 1 0 4
82     // 0 1 1 8
83     // 1 0 0 16
84     // 1 0 1 32
85     // 1 1 0 64
86     // 1 1 1 128
87
88     // kanal przetwornika
89     ADMUX = 0;

```

```

90 }
91
92 void menu(int m) { //obsługa menu programu
93     lcd_clrscr();
94     switch (m) {
95         case 0 :
96             lcd_gotoxy(0,0);
97             lcd_puts("wyp PWM: ");
98             lcd_gotoxy(12,0);
99             lcd_putd(params[m]);
100            return;
101        case 1 :
102            lcd_gotoxy(0,0);
103            lcd_puts("adcvalue: ");
104            lcd_gotoxy(12,0);
105            lcd_putd(adcvalue);
106            return;
107        case 2 :
108            lcd_gotoxy(0,0);
109            lcd_puts("wejście: ");
110            lcd_gotoxy(12,0);
111            lcd_puti(TCCRIA);
112            return;
113    }
114 }
115
116 int main(void) {
117     // inicjalizacja peryferiow
118     lcd_init(LCD_DISP_ON); // wyswietlacz
119     adc_init(); // przetwornika A/C
120     pwm_init(); // wyjścia PWM
121
122     // konfiguracja portow
123     DDRC = _BV(2) | _BV(3) | _BV(4) | _BV(5);
124     PORTC = 0x00;
125     DDRD = 0xff;
126     PORTD = 0x00;
127
128     // deklaracja i inicjalizacja zmiennych
129     char key, key_;
130     params[0]=512;
131     uint8_t refresh = 0, menuitem = 0;
132     uint16_t value = 0, keycount = 0, softdelay = 0, adcvalue_ = 0;
133
134     sei(); // włącz obsługę przerwan
135     menu(menuitem); // wyswietl menu
136
137     while(1) {
138
139         key_ = key;
140         key = kbb_read();
141
142         if (key == key_ && key != 0) {
143             keycount++;
144
145             if (keycount >= 1500) {
146                 refresh = 1;
147                 keycount = 0;
148                 value = 16;

```

```
149     }
150   }
151   else {
152     keycount = 0;
153     value = 1;
154   }
155
156
157   if ( (key && key != key_) || (refresh) ) {
158
159     if (key == 1) { menuitem--;}
160     if (key == 2) { menuitem++;}
161
162     if (key == 3) { params[menuitem]-=value;}
163     if (key == 4) { params[menuitem]+=value;}
164
165     menuitem = menuitem % 3;
166     menu(menuitem);
167     refresh = 0;
168   }
169
170
171   if (99 == softdelay%100) {
172     ADCSRA |= _BV(ADSC);
173   }
174
175   if (softdelay == 2000) {
176     //refresh = 1;
177     softdelay = 0;
178   }
179
180   PWM_out(params[0]);
181   OCR1B=adcvalue;
182
183   softdelay++;
184 }
185 }
```

Dodatek B

Zawartość załączonej płyty CD

Wykaz katalogów wraz zawartością:

- **dokumenty** - dokumenty i specyfikacje wykorzystane przy realizacji pracy
 - AVR
 - avrlibc
 - InTouch
 - Karty Katalogowe
 - Karty Katalogowe (Czujniki)
 - LaTeX
 - Siemens
 - sterowanie
- **instalki** - pliki instalacyjne omawianych programów
- **praca** - źródła pracy w formacie \LaTeX
- **projekty** - projekty wykonane w trakcie realizacji pracy
 - AVR Studio
 - Eagle
 - InTouch
 - kontrollerlab
 - MATLAB
 - OrCAD
 - S7-200
 - TOPServer
- **zrodla** - źródła programów