Sprawozdanie Cw1

Jan Bronicki 249011

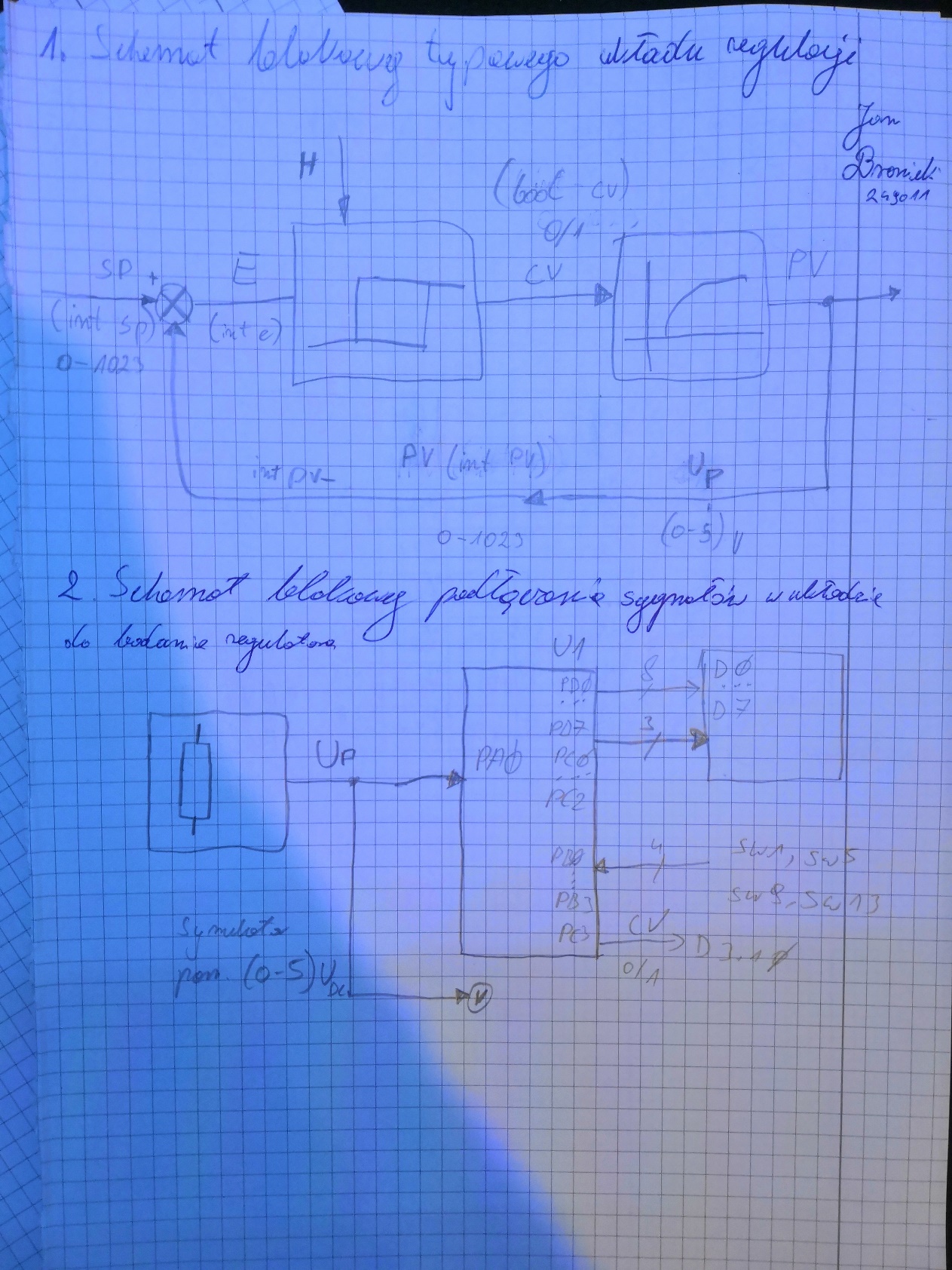
Borys Staszczak 248958

**Temat: Badanie regulatora dwustawnego**

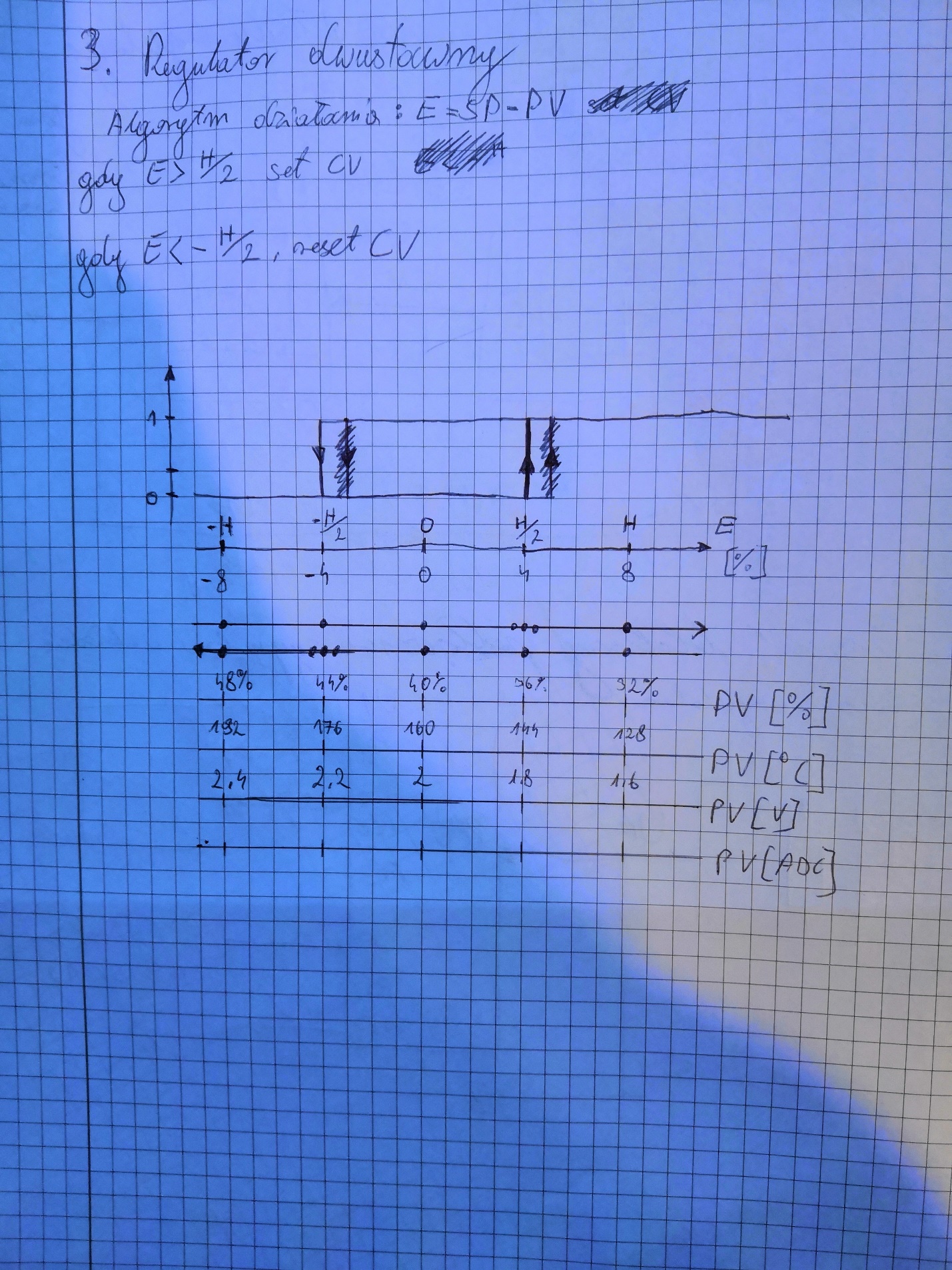
1. **Zadanie do wykonania**

Opracować układ pomiarowy, zmontować układ do badania regulatora, opracować algorytm sterowania w układzie regulacji dwustawnej i przetestować regulator w warunkach laboratoryjnych.

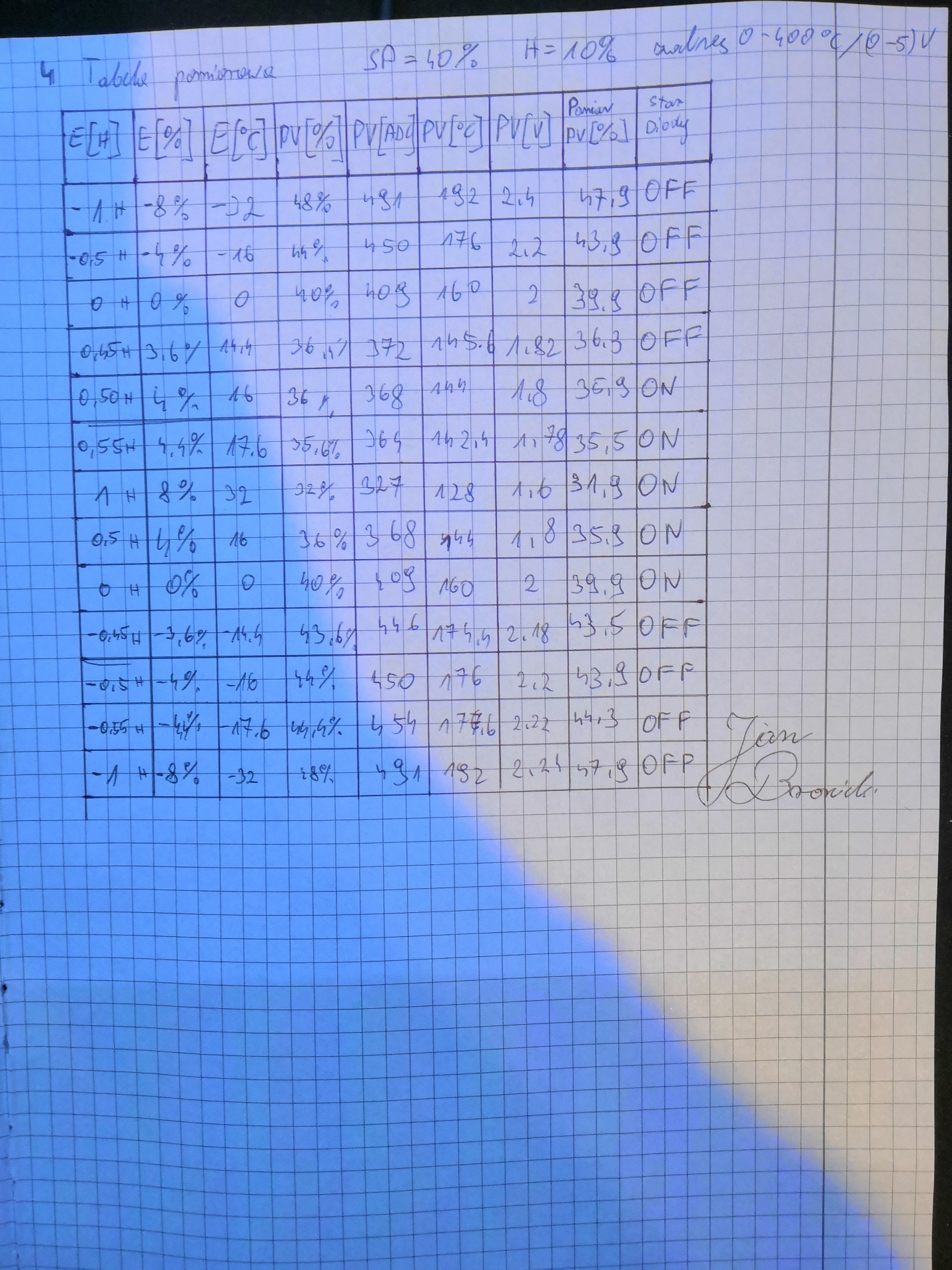
1. **Założenia projektowe**



1. **Regulator dwustawny**



1. **Tabela pomiarowa (każda grupa oblicza dane do tabeli dla „własnych” danych)**



1. **Uwagi i wnioski**
2. **Załącznik nr 1: Kod program**
3. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
4. /\*              ARE 2008              \*/
5. /\*      e-mail: biuro@are.net.pl      \*/
6. /\*      www   : are.net.pl            \*/
7. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
8. // Jan Bronicki 249011
9. // Borys Staszczak 248958
10. #define \_\_AVR\_ATmega32\_\_
11. #define F\_CPU 8000000UL
12. #include <avr/io.h>
13. #include <stdio.h>
14. #include <util/delay.h>
15. #include <string.h>
16. void delay\_ms(int ms)
17. {
18. volatile long unsigned int i;
19. for (i = 0; i < ms; i++)
20. \_delay\_ms(1);
21. }
22. void delay\_us(int us)
23. {
24. volatile long unsigned int i;
25. for (i = 0; i < us; i++)
26. \_delay\_us(1);
27. }
28. #define RS 0
29. #define RW 1
30. #define E 2
31. void LCD2x16\_init(void)
32. {
33. PORTC &= ~(1 << RS);
34. PORTC &= ~(1 << RW);
35. PORTC |= (1 << E);
36. PORTD = 0x38; // dwie linie, 5x7 punktow
37. PORTC &= ~(1 << E);
38. \_delay\_us(120);
39. PORTC |= (1 << E);
40. PORTD = 0x0e; // wlacz wyswietlacz, kursor, miganie
41. PORTC &= ~(1 << E);
42. \_delay\_us(120);
43. PORTC |= (1 << E);
44. PORTD = 0x06;
45. PORTC &= ~(1 << E);
46. \_delay\_us(120);
47. }
48. void LCD2x16\_clear(void)
49. {
50. PORTC &= ~(1 << RS);
51. PORTC &= ~(1 << RW);
52. PORTC |= (1 << E);
53. PORTD = 0x01;
54. PORTC &= ~(1 << E);
55. delay\_ms(120);
56. }
57. void LCD2x16\_putchar(int data)
58. {
59. PORTC |= (1 << RS);
60. PORTC &= ~(1 << RW);
61. PORTC |= (1 << E);
62. PORTD = data;
63. PORTC &= ~(1 << E);
64. \_delay\_us(120);
65. }
66. void LCD2x16\_pos(int wiersz, int kolumna)
67. {
68. PORTC &= ~(1 << RS);
69. PORTC &= ~(1 << RW);
70. PORTC |= (1 << E);
71. delay\_ms(1);
72. PORTD = 0x80 + (wiersz - 1) \* 0x40 + (kolumna - 1);
73. delay\_ms(1);
74. PORTC &= ~(1 << E);
75. \_delay\_us(120);
76. }
77. // Set point (in %)
78. int set\_point = 40;
79. // Histereza (in %)
80. int \_h = 8;
81. // Error value
82. int \_e;
83. // Integer part of the error
84. int int\_e;
85. // Decimal value of the error
86. int dec\_e;
87. // Whole process value (in 0-1023 range)
88. float process\_value;
89. // Process value with decimal part
90. int full\_process\_value;
91. // Integer part of process value
92. int int\_process\_value;
93. // Decimal part of process value
94. int dec\_process\_value;
95. int main(void)
96. {
97. char tmp[16];
98. int i;
99. DDRD = 0xff;
100. PORTD = 0x00;
101. DDRC = 0xff;
102. PORTC = 0x00;
103. DDRB = 0x00;
104. PORTB = 0xff;
105. \_delay\_ms(500);
106. LCD2x16\_init();
107. LCD2x16\_clear();
108. ADMUX = 0x40;
109. ADCSRA = 0xe0;
110. while (1)
111. {
112. // Start an ADC conversion by setting ADSC bit (bit 6)
113. ADCSRA = ADCSRA | (1 << ADSC);
114. // Wait until the ADSC bit has been cleared
115. while (ADCSRA & (1 << ADSC))
116. ;
117. process\_value = ADC;
118. full\_process\_value = (process\_value / 1023.0) \* 1000;
119. int\_process\_value = full\_process\_value / 10;
120. dec\_process\_value = full\_process\_value % 10;
121. \_e = (set\_point \* 10) - full\_process\_value;
122. int\_e = \_e / 10;
123. dec\_e = \_e % 10;
124. // LED On
125. if (\_e > (\_h / 2))
126. {
127. PORTC = ~(0x01 << 5);
128. }
129. // LED Off
130. if (\_e < -(\_h / 2))
131. {
132. PORTC = (0x01 << 5);
133. }
134. if (!(PINB & (8 << PB0)))
135. {
136. set\_point = 50;
137. }
138. if (!(PINB & (4 << PB0)))
139. {
140. set\_point = 40;
141. }
142. if (!(PINB & (2 << PB0)))
143. {
144. \_h = 8;
145. }
146. if (!(PINB & (1 << PB0)))
147. {
148. \_h = 10;
149. }
150. LCD2x16\_pos(1, 1);
151. sprintf(tmp, "SP=%2d PV=%3d.%1d%% ", set\_point, int\_process\_value, abs(dec\_process\_value));
152. for (i = 0; i < 16; i++)
153. {
154. LCD2x16\_putchar(tmp[i]);
155. }
156. LCD2x16\_pos(2, 1);
157. sprintf(tmp, "H=%2d   E=%3d.%1d%%  ", \_h, int\_e, abs(dec\_e));
158. for (i = 0; i < 16; i++)
159. {
160. LCD2x16\_putchar(tmp[i]);
161. }
162. delay\_ms(500);
163. }
164. return 0;
165. }