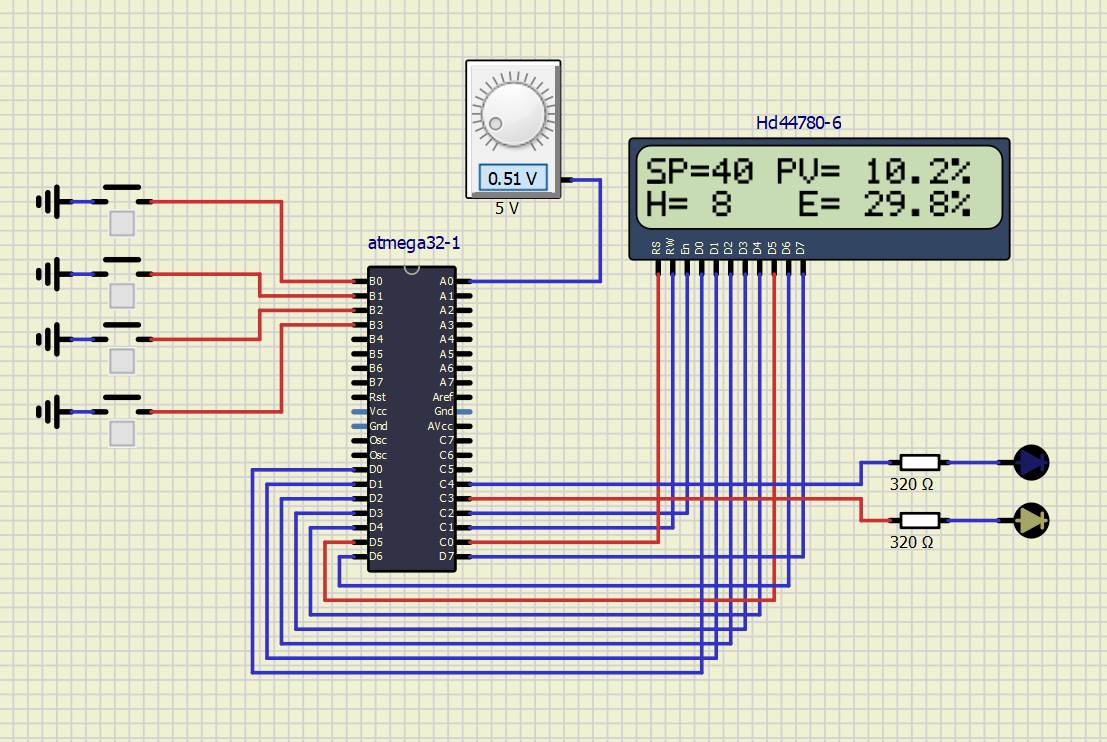
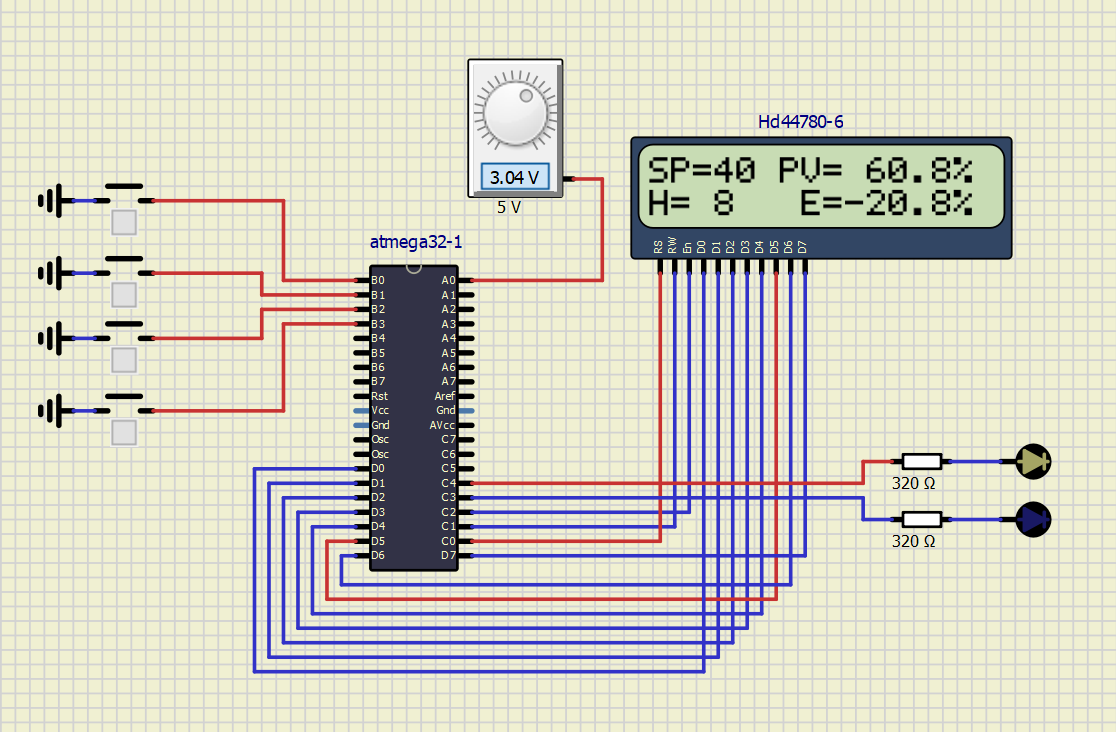
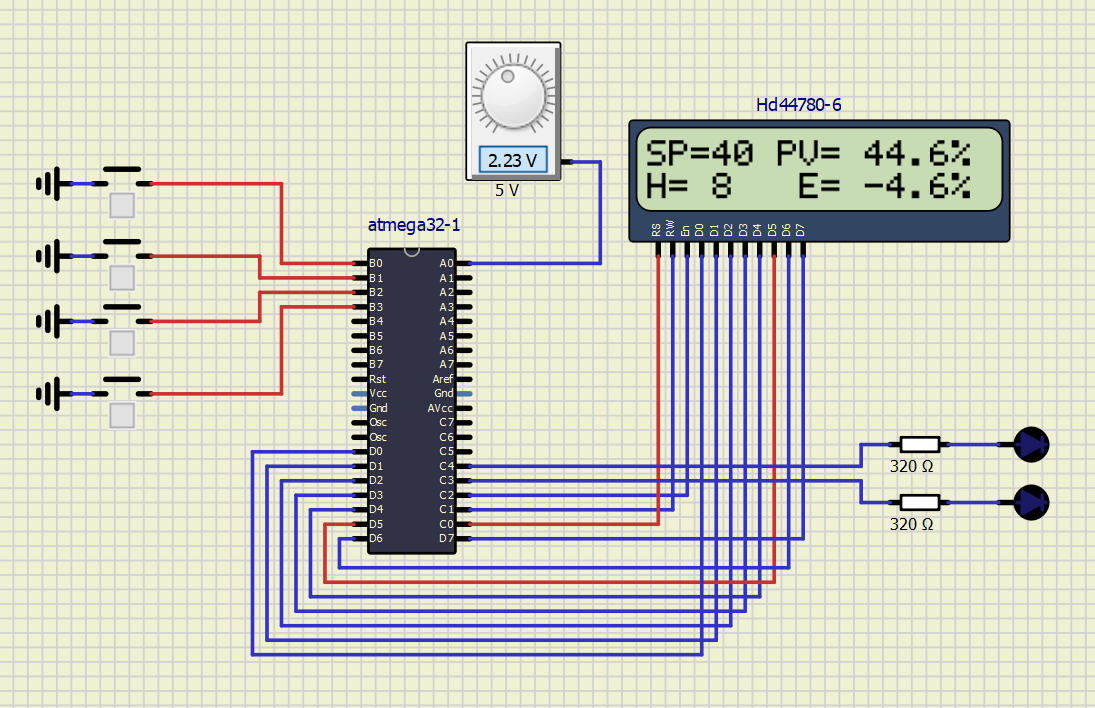


SYMULACJA:







KOD:

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*/\*              ARE 2008              \*/*

*/\*      e-mail: biuro@are.net.pl      \*/*

*/\*      www   : are.net.pl            \*/*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*// Jan Bronicki 249011*

*// Borys Staszczak 248958*

#define \_\_AVR\_ATmega32\_\_

#define F\_CPU 8000000UL

#include <avr/io.h>

#include <stdio.h>

#include <util/delay.h>

#include <string.h>

void delay\_ms(int ms)

{

    volatile long unsigned int i;

    for (i = 0; i < ms; i++)

        \_delay\_ms(1);

}

void delay\_us(int us)

{

    volatile long unsigned int i;

    for (i = 0; i < us; i++)

        \_delay\_us(1);

}

#define RS 0

#define RW 1

#define E 2

void LCD2x16\_init(void)

{

    PORTC &= ~(1 << RS);

    PORTC &= ~(1 << RW);

    PORTC |= (1 << E);

    PORTD = 0x38; *// dwie linie, 5x7 punktow*

    PORTC &= ~(1 << E);

    \_delay\_us(120);

    PORTC |= (1 << E);

    PORTD = 0x0e; *// wlacz wyswietlacz, kursor, miganie*

    PORTC &= ~(1 << E);

    \_delay\_us(120);

    PORTC |= (1 << E);

    PORTD = 0x06;

    PORTC &= ~(1 << E);

    \_delay\_us(120);

}

void LCD2x16\_clear(void)

{

    PORTC &= ~(1 << RS);

    PORTC &= ~(1 << RW);

    PORTC |= (1 << E);

    PORTD = 0x01;

    PORTC &= ~(1 << E);

    delay\_ms(120);

}

void LCD2x16\_putchar(int data)

{

    PORTC |= (1 << RS);

    PORTC &= ~(1 << RW);

    PORTC |= (1 << E);

    PORTD = data;

    PORTC &= ~(1 << E);

    \_delay\_us(120);

}

void LCD2x16\_pos(int wiersz, int kolumna)

{

    PORTC &= ~(1 << RS);

    PORTC &= ~(1 << RW);

    PORTC |= (1 << E);

    delay\_ms(1);

    PORTD = 0x80 + (wiersz - 1) \* 0x40 + (kolumna - 1);

    delay\_ms(1);

    PORTC &= ~(1 << E);

    \_delay\_us(120);

}

*// Set point (in 0.1%)*

int \_sp = 400;

*// Histereza (in 0.1%)*

int \_h = 80;

*// Nieczułość (in 0.1%)*

int \_n = 160;

*// Error value*

int \_e;

*// Integer part of the error*

int int\_e;

*// Decimal value of the error*

int dec\_e;

*// Whole process value (in 0-1023 range)*

float process\_value;

*// Process value with decimal part*

int \_pv;

*// Integer part of process value*

int \_ipv;

*// Decimal part of process value*

int \_decpv;

int main(void)

{

    char tmp[16];

    int i;

    DDRD = 0xff;

    PORTD = 0x00;

    DDRC = 0xff;

    PORTC = 0x00;

    DDRB = 0x00;

    PORTB = 0xff;

    \_delay\_ms(500);

    LCD2x16\_init();

    LCD2x16\_clear();

    ADMUX = 0x40;

    ADCSRA = 0xe0;

    while (1)

    {

*// Start an ADC conversion by setting ADSC bit (bit 6)*

        ADCSRA = ADCSRA | (1 << ADSC);

*// Wait until the ADSC bit has been cleared*

        while (ADCSRA & (1 << ADSC))

            ;

*//\_n=\_h+\_h;*

        process\_value = ADC;

        \_pv = (process\_value / 1023.0) \* 1000;

        \_ipv = \_pv / 10;

        \_decpv = \_pv % 10;

        \_e = \_sp - \_pv;

        int\_e = \_e / 10;

        dec\_e = \_e % 10;

*// LED CV1 ON*

        if (\_e > ((\_n/2)+\_h))

        {

            PORTC = ~(0x01 << 4);

        }

*// LED OFF*

        if(\_e < \_n/2 && \_e > -\_n/2)

        {

            PORTC=(0x00);

        }

*// LED CV2 ON*

        if (\_e < ((-\_n/2)-\_h))

        {

            PORTC = ~(0x01 << 3);

        }

        if (!(PINB & (8 << PB0)))

        {

            \_sp = 50;

        }

        if (!(PINB & (4 << PB0)))

        {

            \_sp = 40;

        }

        if (!(PINB & (2 << PB0)))

        {

            \_h = 8;

            \_n = 16;

        }

        if (!(PINB & (1 << PB0)))

        {

            \_h = 10;

            \_n = 20;

        }

        LCD2x16\_pos(1, 1);

        sprintf(tmp, "SP=%2d PV=%3d.%1d%% ", \_sp/10, \_ipv, abs(\_decpv));

        for (i = 0; i < 16; i++)

        {

            LCD2x16\_putchar(tmp[i]);

        }

        LCD2x16\_pos(2, 1);

        sprintf(tmp, "H=%2d   E=%3d.%1d%% ", \_h/10, int\_e, abs(dec\_e));

        for (i = 0; i < 16; i++)

        {

            LCD2x16\_putchar(tmp[i]);

        }

        delay\_ms(500);

    }

    return 0;

}