

```

#include <inttypes.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

#define XTAL          1000000L
#define TIMER_CLOCK    2
#define blue OCR0A           //nazwę rejestru
OCR0A zastąp nazwą "blue" (wyjście PB2))
#define red OCR0B           //nazwę rejestru
OCR0B zastąp nazwą "red" (wyjście PD5)
#define green OCR1A        //nazwę rejestru
OCR1A zastąp nazwą "green" (wyjście PB3)

static volatile uint8_t i=1;           //deklaracja
zmiennej globalnej "i"
volatile uint8_t b;                   //deklaracja
zmiennej globalnej "b"

/*****KONFIGURACJA REJESTRÓW I WE/WY*****/
init_setup()                          //podprogram który
jest uruchamiany                      //w programie
{
    głównym, konfiguruje rejestry i porty
    DDRB|=(1<<PB2)|(1<<PB3);          //PB2 i PB3 jako
    wyjścia                          //PD5 jako wyjście,
    DDRD|=(1<<PD5)|(0<<PD0);          //PD0 podciągnięte
    PD0 jako wejście                //do VCC
    PORTD|=(1<<PD0);

    //opisane na samym dole programu

    TCCR0A|=(1<<WGM00)|(0<<WGM01)|(0<<WGM02)|(1<<COM0A1)|(1<<COM0A0)|(1<<COM0B1)|(1<<COM0B0);
    TCCR0B|=(1<<CS00);

    TCCR1A|=(1<<WGM10)|(0<<WGM11)|(0<<WGM12)|(1<<COM1A1)|(1<<COM1A0)|(1<<COM1B1)|(1<<COM1B0);
    TCCR1B|=(1<<CS10)|(1<<CS11)|(1<<WGM12);

    OCR1A = 3600;                    //wpisanie do
    rejestru OCR1A stałej wartości   //od niej zależy
    jak często występuje przerwanie //ustawienie OCIE1B w
    TIMSK|=(1<<OCIE1B);              //rejestrze TIMSk odblokowuje przerwanie

    sei();                           //globalne
    zezwolenie na przerwania
}

/*****OBSŁUGA PRZERWANIA*****/
ISR(TIMER1_COMPB_vect)               //przerwanie od
timera_1 wykonywane jest            //gdy wartość
{
    timera_1 zrówna się z wartością OCR1A //jeżeli PD0
    if(bit_is_clear(PIND,0))           //wciśnięty
    {
        _delay_ms(80);                //opóźnienie 80ms
        incr_i();                      //idź do
    }
    podprogramu "incr_i"
}

/*****PODPROGRAM ZWIĘKSZAJĄCY ZMIENNĄ "i"*****/
void incr_i()                        //podprogram
obsługujący zmienną "i"
{
    _delay_ms(80);                    //opóźnienie 80ms
    while(bit_is_clear(PIND,0)){      //nic nie rób gdy
    wciśnięty PD0
    }
}

```

```

    _delay_ms(80); //opóźnienie 80ms
    i=i+1; //zwiększ wartość
    zmiennej "i" o 1
    _delay_ms(100); //opóźnienie 100ms
    if(i==5) i=1; //jeżeli i=5 to i=1
}
/*****
/*****PODPROGRAMY DLA TRYBÓW OŚWIEPLENIA*****/
void red_mode() //podprogram który
na stałe wpisuje //wartości do
{
    rejestrów OCR0A, OCR0B, OCR1A
    red=0;
    blue=255;
    green=255;
}

void blue_mode()
{
    blue=0;
    red=255;
    green=255;
}

void green_mode()
{
    green=0;
    blue=255;
    red=255;
}

void pwm_mode() //podprogram
płynnie zmieniający świecenie diod
{
    for(b=0;b<255;b++)
    {
        red=b; //do rejestru OCR0B
        _delay_ms(10);
    }
    for(b=255;b>0;b--)
    {
        red=b;
        _delay_ms(10);
    }
    for(b=0;b<255;b++)
    {
        blue=b;
        _delay_ms(10);
    }
    for(b=255;b>0;b--)
    {
        blue=b;
        _delay_ms(10);
    }
    for(b=0;b<255;b++)
    {
        green=b;
        _delay_ms(10);
    }
    for(b=255;b>0;b--)
    {
        green=b;
        _delay_ms(10);
    }
}

int main(void) //program główny
{
    init_setup(); //wywołanie
    podprogramu "init_setup"
    while(1) //nieskończona pętla
    {

```

```
if(i==1) pwm_mode(); //jeżeli i=1
wywołaj podprogram "pwm_mode"
if(i==2) red_mode();
if(i==3) blue_mode();
if(i==4) green_mode();

}

//(1<<WGM00)|(0<<WGM01)|(0<<WGM02) - PWM phase correct, tryb pracy PWM
//(1<<COM0A1)|(1<<COM0A0) - Set OC0A on Compare Match when up-counting. Clear OC0A on
Compare Match when down-counting.
//(1<<COM0B1)|(1<<COM0B0) - Set OCR0B on Compare Match when up-counting. Clear OCR0B on
Compare Match when down-counting.
//(1<<CS00) - No clock source (Timer/Counter stopped)

//(1<<WGM10)|(0<<WGM11)|(0<<WGM12) - PWM, Phase Correct, 8-bit
//(1<<COM1A1)|(1<<COM1A0) - Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at TOP
//(1<<COM1B1)|(1<<COM1B0) - Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at TOP
//(1<<WGM12) - CTC
//(1<<CS10)|(1<<CS11) - clkI/O/64 (From prescaler)
```