**Listing 1. Plik nagłówkowy mechanizmu multipleksowania wyświetlaczy LED**

#define SEG\_DDR DDRB

#define SEG\_PORT PORTB

//Definicje konfiguracji poszczególnych segmentów (katod)

#define SEG\_A PB7

#define SEG\_B PB0

#define SEG\_C PB2

#define SEG\_D PB4

#define SEG\_E PB5

#define SEG\_F PB6

#define SEG\_G PB1

#define SEG\_DP PB3

//Port katod, jako port wyjściowy

#define SEG\_AS\_OUTPUT SEG\_DDR = 0xFF

//Wszystkie segmenty (katody) wygaszone (aktywny stan "0", gdyż sterujemy bezpośrednio katodami diod LED)

#define SEG\_BLANK SEG\_PORT = 0xFF

#define COM\_DDR DDRD

#define COM\_PORT PORTD

//Definicje konfiguracji poszczególnych wspólnych anod

#define COM\_DIG0 PD6

#define COM\_DIG1 PD4

#define COM\_DIG2 PD5

#define COM\_DIG3 PD3

//Port wspólnych anod, jako port wyjściowy

#define COM\_AS\_OUTPUT COM\_DDR |= (1<<COM\_DIG3)|(1<<COM\_DIG2)|(1<<COM\_DIG1)|(1<<COM\_DIG0)

//Wszystkie wspólne anody wyłączone (aktywny stan "0", gdyż sterujemy bazami tranzystorów PNP)

#define COM\_BLANK COM\_PORT |= (1<<COM\_DIG3)|(1<<COM\_DIG2)|(1<<COM\_DIG1)|(1<<COM\_DIG0)

//Definicja bitu odpowiedzialnego za miganie cyfry

#define BLINKING\_BIT 0b10000000

//Index wygaszonej cyfry w tablicy DIGITS

#define BLANK\_DIGIT\_NR 16

**Listing 2. Niezbędne definicje zmiennych mechanizmu multipleksowania wyświetlaczy LED**

//Definicje cyfr wyświetlacza (aktywny stan "0", gdyż sterujemy bezpośrednio katodami diod LED)

const uint8\_t DIGITS**[**17**]** PROGMEM **=**

**{**

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_B**)|(**1**<<**SEG\_C**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_F**)),** //0

**(**uint8\_t**)** **~((**1**<<**SEG\_B**)|(**1**<<**SEG\_C**)),** //1

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_B**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_G**)),** //2

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_B**)|(**1**<<**SEG\_C**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_G**)),** //3

**(**uint8\_t**)** **~((**1**<<**SEG\_B**)|(**1**<<**SEG\_C**)|(**1**<<**SEG\_F**)|(**1**<<**SEG\_G**)),** //4

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_C**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_F**)|(**1**<<**SEG\_G**)),** //5

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_C**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_F**)|(**1**<<**SEG\_G**)),** //6

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_B**)|(**1**<<**SEG\_C**)),** //7

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_B**)|(**1**<<**SEG\_C**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_F**)|(**1**<<**SEG\_G**)),** //8

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_B**)|(**1**<<**SEG\_C**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_F**)|(**1**<<**SEG\_G**)),** //9

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_B**)|(**1**<<**SEG\_C**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_F**)|(**1**<<**SEG\_G**)),** //A

**(**uint8\_t**)** **~((**1**<<**SEG\_C**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_F**)|(**1**<<**SEG\_G**)),** //b

**(**uint8\_t**)** **~((**1**<<**SEG\_D**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_G**)),** //c

**(**uint8\_t**)** **~((**1**<<**SEG\_B**)|(**1**<<**SEG\_C**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_G**)),** //d

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_D**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_F**)|(**1**<<**SEG\_G**)),** //E

**(**uint8\_t**)** **~((**1**<<**SEG\_A**)|(**1**<<**SEG\_E**)|(**1**<<**SEG\_F**)|(**1**<<**SEG\_G**)),** //F

0xFF //Wyświetlacz wygaszony

**};**

//Definicje dla portu sterującego wspólnymi anodami wyświetlaczy LED (aktywny stan "0", gdyż sterujemy bazami tranzystorów PNP)

const uint8\_t COMS**[**4**]** PROGMEM **=**

**{**

**(**uint8\_t**)** **~(**1**<<**COM\_DIG0**),** //Wspólna anoda cyfry 0 (pierwsza z prawej)

**(**uint8\_t**)** **~(**1**<<**COM\_DIG1**),** //Wspólna anoda cyfry 1

**(**uint8\_t**)** **~(**1**<<**COM\_DIG2**),** //Wspólna anoda cyfry 2

**(**uint8\_t**)** **~(**1**<<**COM\_DIG3**)** //Wspólna anoda cyfry 3 (pierwsza z lewej)

**};**

volatile uint8\_t Digit**[**4**];** //Zmienna przechowująca wartość wyświetlaną na wyświetlaczu LED

volatile uint8\_t dpON**;** //Zmienna odpowiedzialna za wyświetlanie kropki dziesiętnej na pozycji

**Listing 3. Funkcja inicjalizacyjna mechanizmu multipleksowania wyświetlaczy LED**

void initMultiplex**(**void**)**

**{**

//Porty wspólnych anod i katod, jako wyjściowe ze stanami nieaktywnymi na wyjściach

SEG\_BLANK**;**

SEG\_AS\_OUTPUT**;**

COM\_BLANK**;**

COM\_AS\_OUTPUT**;**

//Konfiguracja układu Timer0 w celu generowania przerwania do obsługi multipleksowania wyświetlacza LED (240 Hz)

TCCR0A **=** **(**1**<<**WGM01**);** //Tryb CTC

TCCR0B **=** **(**1**<<**CS01**)|(**1**<<**CS00**);** //Preskaler = 64

OCR0A **=** 64**;** //240 Hz (co 4.167ms)

TIMSK **|=** **(**1**<<**OCIE0A**);** //Uruchomienie przerwania Output Compare Match A

**for(**uint8\_t i**=**0**;** i**<**4**;** **++**i**)** Digit**[**i**]** **=** BLANK\_DIGIT\_NR**;** //Wygaszenie wszystkich cyfr

**}**

**Listing 4. Funkcja obsługi przerwania układu Timer0 odpowiedzialna za realizację mechanizmu multipleksowania wyświetlaczy LED**

ISR**(**TIMER0\_COMPA\_vect**)**

**{**

static uint8\_t Nr**;** //Numer kolejnej cyfry przeznaczonej do wyświetlenia

static uint8\_t timer4ms**;** //Timer programowy 4ms służący do obsługi migania cyfr wyświetlacza LED

register uint8\_t currDigit **=** Digit**[**Nr**];** //Optymalizacja volatile

COM\_BLANK**;** //Wyłączenie wspólnych anod wyświetlaczy LED

**if(**currDigit **&** BLINKING\_BIT**)** //Sprawdzamy, czy dla danej cyfry uruchomiono funkcję migania (ustawiony 7. bit zmiennej)

**{**

**if(++**timer4ms **&** 0x20**)** SEG\_PORT **=** pgm\_read\_byte**(&**DIGITS**[**currDigit **&** **(~**BLINKING\_BIT**)]);** **else** SEG\_BLANK**;** //Migamy

**}**

**else** SEG\_PORT **=** pgm\_read\_byte**(&**DIGITS**[**currDigit **&** **(~**BLINKING\_BIT**)]);** //Pobranie kolejnej cyfry na port katod

**if(**Nr**==**0 **&&** dpON**)** SEG\_PORT **&=** **~(**1**<<**SEG\_DP**);** //Obsługa kropki dziesiętnej na pozycji 0

COM\_PORT **&=** pgm\_read\_byte**(&**COMS**[**Nr**]);** //Włączenie odpowiedniej wspólnej anody (aktywny stan "0")

Nr **=** **(**Nr**+**1**)** **&** 0x03**;**

**}**