**Listing 1. Program multipleksujący wyświetlacz**

**while** **(**1**)** **{**

**for** **(**int i**=**0 **;** i**<**4 **;** i**++){**

setDisplay**(**DSOFF**);**

setDigit**(**i**,**0**);**

setDisplay**(**i**);**

ALT\_USLEEP**(**1000**);**

**}**

**}**

**Listing 2. Funkcja odpowiedzialna za pojedynczy cykl wyświetlania**

void refreshDisplay**(**void**){**

static uint8\_t disp **=** 0**;**

setDisplay**(**DSOFF**);**

setDigit**(**disp**,** 0**);**

setDisplay**(**disp**);**

**if(**disp **<** 3**)** disp**++;**

**else** disp **=** 0**;**

**}**

**Listing 3. Modyfikacja funkcji obsługi przerwań**

void timer0Interrupt**(**void**\*** context**){**

IOWR\_ALTERA\_AVALON\_TIMER\_STATUS**(**TIMER0\_BASE**,** 0**);**

refreshDisplay**();**

**}**

**Listing 4. Przykładowa funkcja wyświetlająca dane**

void setDigit**(**uint8\_t digit**,** uint8\_t dp**,** uint8\_t pos**){**

**if(**pos **<** 4**){**

**if(**digit **>** 16**){** // dla wartości spoza zakresu gasimy wszystkie segmenty

displayData**[**pos**]** **=** 0**|(**dp**!=**0**?(**1**<<**SEG\_DP**):**0**);**

**}else{** // dla pozostałych wartości wyświetlamy odpowiednią liczbę

displayData**[**pos**]** **=** digits**[**digit**]** **|** **(**dp**!=**0**?(**1**<<**SEG\_DP**):**0**);**

**}**

**}**

**}**

void intDisplayDec**(**uint16\_t number**){**

**for(**uint8\_t i **=** 0 **;** i **<** 4 **;** i**++){**

setDigit**(**number **%** 10**,** 0**,** i**);**

number **/=** 10**;**

**}**

**}**