**Listing 1. Procedura sterująca zapaleniem i gaszeniem diod LED modułu**

/\* File Name : hal\_entry.c

\* Description : This is a very simple example application that

\* blinks all LEDs on a board.

\*/

#include "hal\_data.h"

/\* @brief Blinky example application

\* Blinks all leds at a rate of 1 second using the software delay

\* function provided by the BSP. Only references two other modules

\* including the BSP, IOPORT.

\*/

void hal\_entry**(**void**)**

**{**

/\* Define the units to be used with the software delay function \*/

const bsp\_delay\_units\_t bsp\_delay\_units **=** BSP\_DELAY\_UNITS\_MILLISECONDS**;**

/\* Set the blink frequency (must be <= bsp\_delay\_units \*/

const uint32\_t freq\_in\_hz **=** 2**;**

/\* Calculate the delay in terms of bsp\_delay\_units \*/

const uint32\_t delay **=** bsp\_delay\_units**/**freq\_in\_hz**;**

/\* LED type structure \*/

bsp\_leds\_t leds**;**

/\* LED state variable \*/

ioport\_level\_t level **=** IOPORT\_LEVEL\_HIGH**;**

/\* Get LED information for this board \*/

R\_BSP\_LedsGet**(&**leds**);**

/\* If this board has no LEDs then trap here \*/

**if** **(**0 **==** leds**.**led\_count**)**

**{**

**while(**1**);** // There are no LEDs on this board

**}**

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

**{**

/\* Determine the next state of the LEDs \*/

**if(**IOPORT\_LEVEL\_LOW **==** level**)**

**{**

level **=** IOPORT\_LEVEL\_HIGH**;**

**}**

**else**

**{**

level **=** IOPORT\_LEVEL\_LOW**;**

**}**

/\* Update all board LEDs \*/

**for(**uint32\_t i **=** 0**;** i **<** leds**.**led\_count**;** i**++)**

**{**

g\_ioport**.**p\_api**->**pinWrite**(**leds**.**p\_leds**[**i**],** level**);**

**}**

/\* Delay \*/

R\_BSP\_SoftwareDelay**(**delay**,** bsp\_delay\_units**);**

**}**

**}**

**Listing 2. Wsparcie użycia diod LED z pliku bsp\_leds.c**

/\* Private global variables and functions \*/

/\* Array of LED IOPORT pins. \*/

static const ioport\_port\_pin\_t g\_bsp\_prv\_leds**[]** **=**

**{**

IOPORT\_PORT\_03\_PIN\_03**,** ///< LED2

IOPORT\_PORT\_01\_PIN\_13**,** ///< LED3

**};**

/\* Exported global variables (to be accessed by other files) \*/

/\* Structure with LED information for this board. Recommended to get this information through R\_BSP\_LedsGet() function

\* instead of using this structure directly to remove dependency on structure name. \*/

const bsp\_leds\_t g\_bsp\_leds **=**

**{**

**.**led\_count **=** **(**uint16\_t**)(((**uint16\_t**)sizeof(**g\_bsp\_prv\_leds**)** **/** **(**uint16\_t**)sizeof(**g\_bsp\_prv\_leds**[**0U**]))),**

**.**p\_leds **=** **&**g\_bsp\_prv\_leds**[**0**]**

**};**