**Listing 1. Konfiguracja GPIO**

void PmodLED\_Config**(**void**)**

**{**

// Enable clock for all required GPIO ports.

\_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE**();**

\_\_HAL\_RCC\_GPIOB\_CLK\_ENABLE**();**

\_\_HAL\_RCC\_GPIOE\_CLK\_ENABLE**();**

// Initialize all GPIO as outputs.

GPIO\_InitTypeDef GPIO\_InitStruct**;**

GPIO\_InitStruct**.**Mode **=** GPIO\_MODE\_OUTPUT\_PP**;**

GPIO\_InitStruct**.**Pull **=** GPIO\_PULLUP**;**

GPIO\_InitStruct**.**Speed **=** GPIO\_SPEED\_FREQ\_LOW**;**

GPIO\_InitStruct**.**Pin **=** GPIO\_PIN\_1 **|** GPIO\_PIN\_7**;**

HAL\_GPIO\_Init**(**GPIOA**,** **&**GPIO\_InitStruct**);**

GPIO\_InitStruct**.**Pin **=** GPIO\_PIN\_0**;**

HAL\_GPIO\_Init**(**GPIOB**,** **&**GPIO\_InitStruct**);**

GPIO\_InitStruct**.**Pin **=** GPIO\_PIN\_14**;**

HAL\_GPIO\_Init**(**GPIOE**,** **&**GPIO\_InitStruct**);**

// Turn off all LEDs

HAL\_GPIO\_WritePin**(**GPIOB**,** GPIO\_PIN\_0**,** GPIO\_PIN\_RESET**);**

HAL\_GPIO\_WritePin**(**GPIOA**,** GPIO\_PIN\_7**,** GPIO\_PIN\_RESET**);**

HAL\_GPIO\_WritePin**(**GPIOE**,** GPIO\_PIN\_14**,** GPIO\_PIN\_RESET**);**

HAL\_GPIO\_WritePin**(**GPIOA**,** GPIO\_PIN\_1**,** GPIO\_PIN\_RESET**);**

**}**

**Listing 2. Fragment funkcji PmodLED\_ResetLed**

void PmodLED\_ResetLed**(**PmodLED\_Led led**)**

**{**

**switch(**led**)** **{**

**case** PmodLED\_Led0**:**

HAL\_GPIO\_WritePin**(**GPIOB**,** GPIO\_PIN\_0**,** GPIO\_PIN\_RESET**);**

**break;**

**case** PmodLED\_Led1**:**

HAL\_GPIO\_WritePin**(**GPIOA**,** GPIO\_PIN\_7**,** GPIO\_PIN\_RESET**);**

**break;**

**case** PmodLED\_Led2**:**

HAL\_GPIO\_WritePin**(**GPIOE**,** GPIO\_PIN\_14**,** GPIO\_PIN\_RESET**);**

**break;**

**case** PmodLED\_Led3**:**

HAL\_GPIO\_WritePin**(**GPIOA**,** GPIO\_PIN\_1**,** GPIO\_PIN\_RESET**);**

**break;**

**}**

**}**

**Listing 3. Fragment funkcji PmodLED\_SetLed**

void PmodLED\_SetLed**(**PmodLED\_Led led**)**

**{**

**switch(**led**)** **{**

**case** PmodLED\_Led0**:**

HAL\_GPIO\_WritePin**(**GPIOB**,** GPIO\_PIN\_0**,** GPIO\_PIN\_SET**);**

**break;**

**case** PmodLED\_Led1**:**

HAL\_GPIO\_WritePin**(**GPIOA**,** GPIO\_PIN\_7**,** GPIO\_PIN\_SET**);**

**break;**

**case** PmodLED\_Led2**:**

HAL\_GPIO\_WritePin**(**GPIOE**,** GPIO\_PIN\_14**,** GPIO\_PIN\_SET**);**

**break;**

**case** PmodLED\_Led3**:**

HAL\_GPIO\_WritePin**(**GPIOA**,** GPIO\_PIN\_1**,** GPIO\_PIN\_SET**);**

**break;**

**}**

**}**

**Listing 4. Konfiguracja SPI**

void PmodALS\_Config**(**void**)**

**{**

// Configure the SPI connected to the Pmod module. Only the MISO line is required.

pmodAlsSpi**.**Instance **=** SPI1**;**

pmodAlsSpi**.**Init**.**Mode **=** SPI\_MODE\_MASTER**;**

pmodAlsSpi**.**Init**.**Direction **=** SPI\_DIRECTION\_2LINES\_RXONLY**;**

pmodAlsSpi**.**Init**.**DataSize **=** SPI\_DATASIZE\_16BIT**;**

pmodAlsSpi**.**Init**.**CLKPolarity **=** SPI\_POLARITY\_HIGH**;**

pmodAlsSpi**.**Init**.**CLKPhase **=** SPI\_PHASE\_1EDGE**;**

pmodAlsSpi**.**Init**.**NSS **=** SPI\_NSS\_SOFT**;**

pmodAlsSpi**.**Init**.**BaudRatePrescaler **=** SPI\_BAUDRATEPRESCALER\_64**;**

pmodAlsSpi**.**Init**.**FirstBit **=** SPI\_FIRSTBIT\_MSB**;**

pmodAlsSpi**.**Init**.**TIMode **=** SPI\_TIMODE\_DISABLE**;**

pmodAlsSpi**.**Init**.**CRCCalculation **=** SPI\_CRCCALCULATION\_DISABLE**;**

pmodAlsSpi**.**Init**.**NSSPMode **=** SPI\_NSS\_PULSE\_DISABLE**;**

HAL\_SPI\_Init**(&**pmodAlsSpi**);**

**}**

**Listing 5. Konfiguracja zegarów peryferiów i linii GPIO**

void HAL\_SPI\_MspInit**(**SPI\_HandleTypeDef **\***hspi**)**

**{**

// Initialize GPIO used by the SPI1 peripheral. The CS is controlled by the software (PB0 pin).

\_\_HAL\_RCC\_SPI1\_CLK\_ENABLE**();**

\_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE**();**

\_\_HAL\_RCC\_GPIOB\_CLK\_ENABLE**();**

\_\_HAL\_RCC\_GPIOE\_CLK\_ENABLE**();**

GPIO\_InitTypeDef GPIO\_InitStruct**;**

GPIO\_InitStruct**.**Mode **=** GPIO\_MODE\_AF\_PP**;**

GPIO\_InitStruct**.**Pull **=** GPIO\_PULLDOWN**;**

GPIO\_InitStruct**.**Speed **=** GPIO\_SPEED\_FREQ\_VERY\_HIGH**;**

GPIO\_InitStruct**.**Alternate **=** GPIO\_AF5\_SPI1**;**

GPIO\_InitStruct**.**Pin **=** GPIO\_PIN\_1**;**

HAL\_GPIO\_Init**(**GPIOA**,** **&**GPIO\_InitStruct**);**

GPIO\_InitStruct**.**Pin **=** GPIO\_PIN\_14**;**

HAL\_GPIO\_Init**(**GPIOE**,** **&**GPIO\_InitStruct**);**

GPIO\_InitStruct**.**Mode **=** GPIO\_MODE\_OUTPUT\_PP**;**

GPIO\_InitStruct**.**Pull **=** GPIO\_PULLDOWN**;**

GPIO\_InitStruct**.**Pin **=** GPIO\_PIN\_0**;**

HAL\_GPIO\_Init**(**GPIOB**,** **&**GPIO\_InitStruct**);**

HAL\_GPIO\_WritePin**(**GPIOB**,** GPIO\_PIN\_0**,** GPIO\_PIN\_SET**);**

**}**

**Listing 6. Fragment funkcji PmodLED\_SetLed**

uint8\_t PmodALS\_GetValue**(**void**)**

**{**

uint16\_t value **=** 0**;**

HAL\_GPIO\_WritePin**(**GPIOB**,** GPIO\_PIN\_0**,** GPIO\_PIN\_RESET**);**

HAL\_SPI\_Receive**(&**pmodAlsSpi**,** **(**uint8\_t**\*)&**value**,** 1**,** 100**);**

HAL\_GPIO\_WritePin**(**GPIOB**,** GPIO\_PIN\_0**,** GPIO\_PIN\_SET**);**

// The data obtained from the ADC is 16 bit, but the light level value is only 8 bit.

// It is shifted left by 4 bits.

**return** **(**value **>>** 4**);**

**}**