**Listing 1. Konfiguracja interfejsu SPI do komunikacji z modułem PmodCLS**

// Enable clock for all GPIO ports required for Pmod connector configuration.

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

// Configure the SPI peripheral for 2 data lines and Chip Select pin managed by hardware.

pmodClsSpi**.**Instance **=** SPI1**;**

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

pmodClsSpi**.**Init**.**Direction **=** SPI\_DIRECTION\_2LINES**;**

pmodClsSpi**.**Init**.**DataSize **=** SPI\_DATASIZE\_8BIT**;**

pmodClsSpi**.**Init**.**CLKPolarity **=** SPI\_POLARITY\_LOW**;**

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

pmodClsSpi**.**Init**.**NSS **=** SPI\_NSS\_HARD\_OUTPUT**;**

pmodClsSpi**.**Init**.**BaudRatePrescaler **=** SPI\_BAUDRATEPRESCALER\_128**;**

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

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

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

pmodClsSpi**.**Init**.**NSSPMode **=** SPI\_NSS\_PULSE\_ENABLE**;**

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

Listing 2. Konfiuracja pinów GPIO **do** komunikacji SPI z modułem PmodCLS

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 **|** GPIO\_PIN\_7**;**

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

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

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

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

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

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

**Listing 3. Komendy odpowiedzialne na inicjalizację wyświetlacza**

write**(**"\x1b[j"**,** 3**);** // Clear the display and home cursor.

write**(**"\x1b[0h"**,** 4**);** // Set the display mode (16 characters wrap)

write**(**"\x1b[0c"**,** 4**);** // Set cursor mode (cursor off).

write**(**"\x1b[0;0H"**,** 6**);** // Set the cursor position to 0,0.

**Listing 4. Funkcja wypisujaca znaki na wyświetlaczu**

void PmodCLS\_Write**(**uint8\_t line**,** char**\*** text**,** uint32\_t len**)**

**{**

// Verify the arguments according to the defined limits.

**if(**line **>=** MAX\_LINES **||** len **>** MAX\_LINE\_CHARACTERS**)** **return;**

// Set the cursor position to the beginning of the selected line..

**if(**line **==** 0**)** write**(**"\x1b[0;0H"**,** 6**);**

**else** write**(**"\x1b[1;0H"**,** 6**);**

write**(**text**,** len**);**

**}**

**Listing 5. Inicjalizacja interfejsu SPI dla PmodDPG1**

SPI\_HandleTypeDef pmodDpg1Spi**;**

void PmodDPG1\_Config**(**void**)**

**{**

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

// The CS line is controlled by the hardware.

pmodDpg1Spi**.**Instance **=** SPI1**;**

pmodDpg1Spi**.**Init**.**Mode **=** SPI\_MODE\_MASTER**;**

pmodDpg1Spi**.**Init**.**Direction **=** SPI\_DIRECTION\_2LINES\_RXONLY**;**

pmodDpg1Spi**.**Init**.**DataSize **=** SPI\_DATASIZE\_16BIT**;**

pmodDpg1Spi**.**Init**.**CLKPolarity **=** SPI\_POLARITY\_HIGH**;**

pmodDpg1Spi**.**Init**.**CLKPhase **=** SPI\_PHASE\_1EDGE**;**

pmodDpg1Spi**.**Init**.**NSS **=** SPI\_NSS\_HARD\_OUTPUT**;**

pmodDpg1Spi**.**Init**.**BaudRatePrescaler **=** SPI\_BAUDRATEPRESCALER\_64**;**

pmodDpg1Spi**.**Init**.**FirstBit **=** SPI\_FIRSTBIT\_MSB**;**

pmodDpg1Spi**.**Init**.**TIMode **=** SPI\_TIMODE\_DISABLE**;**

pmodDpg1Spi**.**Init**.**CRCCalculation **=** SPI\_CRCCALCULATION\_DISABLE**;**

pmodDpg1Spi**.**Init**.**NSSPMode **=** SPI\_NSS\_PULSE\_ENABLE**;**

HAL\_SPI\_Init**(&**pmodDpg1Spi**);**

**}**

**Listing 6. Inicjalizacja pinów interfejsu SPI dla PmodDPG1**

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

**{**

// Initialize GPIO used by the SPI2 peripheral.

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

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

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

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

GPIO\_InitTypeDef GPIO\_InitStruct**;**

// Configure all required GPIO lines in alternate mode.

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

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

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

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

**}**

Listing 7. Odczyt i konwersja wartości różnicy ciśnienia

uint32\_t PmodDPG1\_GetValue**(**void**)**

**{**

/\* Receive the 12-bit value from the ADC and convert it to

the pressure difference in [Pa]. The conversion equation

is taken from the PmodDPG1 reference manual. \*/

uint16\_t value**[**2**]** **=** **{**0**};**

HAL\_SPI\_Receive**(&**pmodDpg1Spi**,** **(**uint8\_t**\*)**value**,** 2**,** 100**);**

value**[**1**]** **=** **((**value**[**1**]** **\*** 1000 **/** 4096**)** **-** 80**)** **/** 0.09**;**

**return** value**[**1**];**

**}**

**Listing 8. Inicjalizacja interfejsu SPI dla PmodISNS20**

SPI\_HandleTypeDef pmodIsns20Spi**;**

void PmodISNS20\_Config**(**void**)**

**{**

/\* Configure the SPI connected to the Pmod module. Only the

MISO line is required. \*/

pmodIsns20Spi**.**Instance **=** SPI1**;**

pmodIsns20Spi**.**Init**.**Mode **=** SPI\_MODE\_MASTER**;**

pmodIsns20Spi**.**Init**.**Direction **=** SPI\_DIRECTION\_2LINES\_RXONLY**;**

pmodIsns20Spi**.**Init**.**DataSize **=** SPI\_DATASIZE\_16BIT**;**

pmodIsns20Spi**.**Init**.**CLKPolarity **=** SPI\_POLARITY\_HIGH**;**

pmodIsns20Spi**.**Init**.**CLKPhase **=** SPI\_PHASE\_1EDGE**;**

pmodIsns20Spi**.**Init**.**NSS **=** SPI\_NSS\_SOFT**;**

pmodIsns20Spi**.**Init**.**BaudRatePrescaler **=** SPI\_BAUDRATEPRESCALER\_64**;**

pmodIsns20Spi**.**Init**.**FirstBit **=** SPI\_FIRSTBIT\_MSB**;**

pmodIsns20Spi**.**Init**.**TIMode **=** SPI\_TIMODE\_DISABLE**;**

pmodIsns20Spi**.**Init**.**CRCCalculation **=** SPI\_CRCCALCULATION\_DISABLE**;**

pmodIsns20Spi**.**Init**.**NSSPMode **=** SPI\_NSS\_PULSE\_DISABLE**;**

HAL\_SPI\_Init**(&**pmodIsns20Spi**);**

**}**

**Listing 9. Inicjalizacja pinów interfejsu SPI dla PmodISNS20**

/\* Initialize GPIO used by the SPI2 peripheral. The CS is configured

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\_PULLUP**;**

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

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

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

Listing 10. Odczyt i konwersja wartości różnicy ciśnienia

int32\_t PmodISNS20\_GetValue**(**void**)**

**{**

uint16\_t adcValue **=** 0**;**

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

HAL\_SPI\_Receive**(&**pmodIsns20Spi**,** **(**uint8\_t**\*)&**adcValue**,** 1**,** 100**);**

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

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

HAL\_SPI\_Receive**(&**pmodIsns20Spi**,** **(**uint8\_t**\*)&**adcValue**,** 1**,** 100**);**

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

/\* The current value in [mA] is calculated according to the formula

given in the PmodISNS20 reference manual. \*/

//return (int32\_t)(((int16\_t)adcValue - 2048) \* 1000.0 / 89.95);

**}**