**Listing 1. Plik programu głównego *main.c***

/\* USER CODE BEGIN Includes \*/

#include "bmp180.h"

/\* USER CODE END Includes \*/

int main**(**void**)**

**{**

/\* Initialize all configured peripherals \*/

MX\_GPIO\_Init**();**

MX\_I2C1\_Init**();**

MX\_USART2\_UART\_Init**();**

/\* USER CODE BEGIN 2 \*/

bmp\_state bmp **=** bmp\_init**(&**hi2c1**);**

bmp\_read\_compensation\_data**(&**bmp**);**

double temperature**,** pressure**,** altitude**;**

char output**[**50**];**

sprintf**(**output**,** "Temperatura: Cisnienie: Wysokosc: \r\n"**);**

HAL\_UART\_Transmit**(&**huart2**,** output**,** strlen**(**output**),** 100**);**

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

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

**{**

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

bmp\_read\_temp\_and\_pressure**(&**bmp**);**

temperature **=** bmp\_get\_temperature**(&**bmp**);**

pressure **=** bmp\_get\_pressure**(&**bmp**);**

altitude **=** bmp\_get\_altitude**(**pressure**,** 1013.25**);**

sprintf**(**output**,** "%+05.1f \*C %04.0f hPa %+07.2f m n.p.m. \r"**,** temperature**,** pressure**,** altitude**);**

HAL\_UART\_Transmit**(&**huart2**,** output**,** strlen**(**output**),** 100**);**

HAL\_Delay**(**100**);**

**}**

/\* USER CODE END 3 \*/

**}**

**Listing 2. Plik nagłówkowy *bmp180.h***

#include "i2c.h"

**typedef** struct bmp\_state **{**

I2C\_HandleTypeDef **\*** i2c**;**

int16\_t AC1**,** AC2**,** AC3**,** B1**,** B2**,** MB**,** MC**,** MD**;**

uint16\_t AC4**,** AC5**,** AC6**;**

int32\_t UT**,** UP**,** B5**;**

**}** bmp\_state**;**

bmp\_state bmp\_init**(**I2C\_HandleTypeDef **\*** i2c**);**

uint8\_t bmp\_read\_data**(**bmp\_state **\*** state**,** uint8\_t reg**);**

void bmp\_write\_data**(**bmp\_state **\*** state**,** uint8\_t reg**,** uint8\_t value**);**

void bmp\_read\_compensation\_data**(**bmp\_state **\*** state**);**

void bmp\_read\_temp\_and\_pressure**(**bmp\_state **\*** state**);**

double bmp\_get\_temperature**(**bmp\_state **\*** state**);**

double bmp\_get\_pressure**(**bmp\_state **\*** state**);**

double bmp\_get\_altitude**(**double p**,** double p0**);**

**Listing 3. Plik biblioteki *bmp180.c***

#include "bmp180.h"

bmp\_state bmp\_init**(**I2C\_HandleTypeDef **\*** i2c**)**

**{**

bmp\_state state**;**

state**.**i2c **=** i2c**;**

state**.**AC1 **=** 0**;**

state**.**AC2 **=** 0**;**

state**.**AC3 **=** 0**;**

state**.**AC4 **=** 0**;**

state**.**AC5 **=** 0**;**

state**.**AC6 **=** 0**;**

state**.**B1 **=** 0**;**

state**.**B2 **=** 0**;**

state**.**MB **=** 0**;**

state**.**MC **=** 0**;**

state**.**MD **=** 0**;**

state**.**UT **=** 0**;**

state**.**UP **=** 0**;**

state**.**B5 **=** 0.0**;**

**return** state**;**

**}**

uint8\_t bmp\_read\_data**(**bmp\_state **\*** state**,** uint8\_t reg**)**

**{**

uint8\_t tmp **=** 0**;**

HAL\_I2C\_Mem\_Read**(**state**->**i2c**,** 0xEF**,** reg**,** 1**,** **&**tmp**,** 1**,** 100**);**

**return** tmp**;**

**}**

void bmp\_write\_data**(**bmp\_state **\*** state**,** uint8\_t reg**,** uint8\_t value**)**

**{**

HAL\_I2C\_Mem\_Write**(**state**->**i2c**,** 0xEE**,** reg**,** 1**,** **&**value**,** 1**,** 100**);**

**}**

void bmp\_read\_compensation\_data**(**bmp\_state **\*** state**)**

**{**

state**->**AC1 **=** **(**bmp\_read\_data**(**state**,** 0xAA**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xAB**);**

state**->**AC2 **=** **(**bmp\_read\_data**(**state**,** 0xAC**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xAD**);**

state**->**AC3 **=** **(**bmp\_read\_data**(**state**,** 0xAE**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xAF**);**

state**->**AC4 **=** **(**bmp\_read\_data**(**state**,** 0xB0**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xB1**);**

state**->**AC5 **=** **(**bmp\_read\_data**(**state**,** 0xB2**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xB3**);**

state**->**AC6 **=** **(**bmp\_read\_data**(**state**,** 0xB4**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xB5**);**

state**->**B1 **=** **(**bmp\_read\_data**(**state**,** 0xB6**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xB7**);**

state**->**B2 **=** **(**bmp\_read\_data**(**state**,** 0xB8**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xB9**);**

state**->**MB **=** **(**bmp\_read\_data**(**state**,** 0xBA**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xBB**);**

state**->**MC **=** **(**bmp\_read\_data**(**state**,** 0xBC**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xBD**);**

state**->**MD **=** **(**bmp\_read\_data**(**state**,** 0xBE**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xBF**);**

**}**

void bmp\_read\_temp\_and\_pressure**(**bmp\_state **\*** state**)**

**{**

bmp\_write\_data**(**state**,** 0xF4**,** 0x2E**);**

HAL\_Delay**(**5**);**

state**->**UT **=** **(**bmp\_read\_data**(**state**,** 0xF6**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xF7**);**

uint8\_t OSS **=** 3**;** // "ultra high resolution"

bmp\_write\_data**(**state**,** 0xF4**,** 0x34 **+** **(**OSS **<<** 6**));**

**if(**OSS **==** 0**)** HAL\_Delay**(**5**);**

**else** **if(**OSS **==** 1**)** HAL\_Delay**(**8**);**

**else** **if(**OSS **==** 2**)** HAL\_Delay**(**14**);**

**else** HAL\_Delay**(**26**);**

state**->**UP **=** **((**bmp\_read\_data**(**state**,** 0xF6**)** **<<** 16**)** **+** **(**bmp\_read\_data**(**state**,** 0xF7**)** **<<** 8**)** **+** bmp\_read\_data**(**state**,** 0xF8**))** **>>** 8**-**OSS**;**

**}**

double bmp\_get\_temperature**(**bmp\_state **\*** state**)**

**{**

int32\_t X1**,** X2**,** T**;**

X1 **=** **(**state**->**UT **-** state**->**AC6**)** **\*** state**->**AC5 **/** 32768**;**

X2 **=** state**->**MC **\*** 2048 **/** **(**X1 **+** state**->**MD**);**

state**->**B5 **=** X1 **+** X2**;**

T **=** **(**state**->**B5 **+** 8**)** **/** 16.0**;**

**return** T **/** 10.0**;**

**}**

double bmp\_get\_pressure**(**bmp\_state **\*** state**)**

**{**

int32\_t B6**,** X1**,** X2**,** X3**,** B3**,** P**;**

uint32\_t B4**,** B7**;**

uint8\_t OSS **=** 3**;**

B6 **=** state**->**B5 **-** 4000**;**

X1 **=** **(**state**->**B2 **\*** **(**B6 **\*** B6 **/** 4096**))** **/** 2048**;**

X2 **=** state**->**AC2 **\*** B6 **/** 2048**;**

X3 **=** X1 **+** X2**;**

B3 **=** **(((**state**->**AC1 **\*** 4 **+** X3**)** **<<** OSS**)** **+** 2**)** **/** 4**;**

X1 **=** state**->**AC3 **\*** B6 **/** 8192**;**

X2 **=** **(**state**->**B1 **\*** **(**B6 **\*** B6 **/** 4096**))** **/** 65536**;**

X3 **=** **((**X1 **+** X2**)** **+** 2**)** **/** 4**;**

B4 **=** state**->**AC4 **\*** **(**uint32\_t**)(**X3 **+** 32768**)** **/** 32768**;**

B7 **=** **((**uint32\_t**)**state**->**UP **-** B3**)** **\*** **(**50000 **>>** OSS**);**

**if(**B7 **<** 0x80000000**)** P **=** **(**B7 **\*** 2**)** **/** B4**;** **else** P **=** **(**B7 **/** B4**)** **\*** 2**;**

X1 **=** **(**P **/** 256**)** **\*** **(**P **/** 256**);**

X1 **=** **(**X1 **\*** 3038**)** **/** 65536**;**

X2 **=** **(-**7357 **\*** P**)** **/** 65536**;**

P **=** P **+** **(**X1 **+** X2 **+** 3791**)** **/** 16**;**

**return** P **/** 100.0**;**

**}**

double bmp\_get\_altitude**(**double p**,** double p0**)**

**{**

**return** 44330 **\*** **(**1 **-** pow**((**p**/**p0**),** **(**1**/**5.255**)));**

**}**