**Listing 1. Plik nagłówkowy *display.h***

#ifndef display\_header

#define display\_header

#include "stm32f4xx\_hal.h"

#include "spi.h"

#include "gpio.h"

enum display\_direction **{**horizontal**,** vertical**};**

struct display\_config

**{**

SPI\_HandleTypeDef **\*** spi**;**

GPIO\_TypeDef **\*** reset\_port**;** uint16\_t reset\_pin**;**

GPIO\_TypeDef **\*** bl\_port**;** uint16\_t bl\_pin**;**

GPIO\_TypeDef **\*** dc\_port**;** uint16\_t ce\_pin**;**

GPIO\_TypeDef **\*** ce\_port**;** uint16\_t dc\_pin**;**

**};**

void display\_setup**(**struct display\_config **\*** cfg**);**

void display\_set\_dxy**(**struct display\_config **\*** cfg**,**

enum display\_direction direction**,**

uint8\_t column**,** uint8\_t row**);**

void display\_write\_cmd**(**struct display\_config **\*** cfg**,** uint8\_t cmd**);**

void display\_write\_data**(**struct display\_config **\*** cfg**,** uint8\_t data**);**

#endif

**Listing 2. Plik biblioteki funkcji *display.c***

#include "display.h"

void display\_setup**(**struct display\_config **\*** cfg**)**

**{**

// Sprzętowy reset wyświetlacza

HAL\_GPIO\_WritePin**(**cfg**->**reset\_port**,** cfg**->**reset\_pin**,** GPIO\_PIN\_RESET**);**

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

HAL\_GPIO\_WritePin**(**cfg**->**reset\_port**,** cfg**->**reset\_pin**,** GPIO\_PIN\_SET**);**

// Włączenie podświetlenia

HAL\_GPIO\_WritePin**(**cfg**->**bl\_port**,** cfg**->**bl\_pin**,** GPIO\_PIN\_SET**);**

// Przejście w tryb poleceń rozszerzonych

// 0x00100001 (0x21) - polecenie

display\_write\_cmd**(**cfg**,** 0x21**);**

// Temperature Coefficient

// 0b00000100 (0x04) - polecenie | 0b00000000 (0x00) - wartość

// = 0b00000100 (0x04)

display\_write\_cmd**(**cfg**,** 0x04**);**

// Bias System

// 0b00010000 (0x10) - polecenie | 0x00000100 (0x04) - wartość

// = 0b00010100 (0x14)

display\_write\_cmd**(**cfg**,** 0x10 **|** 0x04**);**

// Ustawienie napięcia sygnału sterującego matrycą LCD / kontrastu

// 0b10000000 (0x80) - polecenie | 0x01001000 (0x38) - wartość

// = 0x11001000 (0xB8)

display\_write\_cmd**(**cfg**,** 0x80 **|** 0x38**);**

// Powrót do trybu poleceń podstawowych

// 0x00100001 (0x20) - polecenie

display\_write\_cmd**(**cfg**,** 0x20**);**

// Ustawienie trybu pracy wyświetlacza - normalnego

// 0x00001000 (0x08) - polecenie | 0b00000100 (0x04) - wartość

// = 0b00001100 (0x0C)

// Pozostałe tryby pracy: cały wyłączony - 0b00000000 (0x00);

// cały zapalony - 0x00000001 (0x01); odwrócony - 0x00000101 (0x05)

display\_write\_cmd**(**cfg**,** 0x08 **|** 0x04**);**

// Czyszczenie wyświetlacza

**for** **(**int i **=** 0**;** i **<** 504**;** i**++)**

display\_write\_data**(**cfg**,** 0x00**);**

**}**

void display\_set\_dxy**(**struct display\_config **\*** cfg**,**

enum display\_direction direction**,**

uint8\_t column**,** uint8\_t row**)**

**{**

// Przejście w tryb poleceń podstawowych i ustawienie kierunku rysowania

**if(**direction **==** vertical**)**

display\_write\_cmd**(**cfg**,** 0x20 **|** 0x02**);**

**else**

display\_write\_cmd**(**cfg**,** 0x20 **|** 0x00**);**

display\_write\_cmd**(**cfg**,** 0x80 **|** column**);** // Wybór kolumny

display\_write\_cmd**(**cfg**,** 0x40 **|** row**);** // Wybór wiersza

**}**

void display\_write\_cmd**(**struct display\_config **\*** cfg**,** uint8\_t cmd**)** **{**

// Wybór tryby transmisji poleceń

HAL\_GPIO\_WritePin**(**cfg**->**dc\_port**,** cfg**->**dc\_pin**,** GPIO\_PIN\_RESET**);**

// Odblokowanie wejścia zegarowego

HAL\_GPIO\_WritePin**(**cfg**->**ce\_port**,** cfg**->**ce\_pin**,** GPIO\_PIN\_RESET**);**

// Transmisja danych interfejsem SPI

HAL\_SPI\_Transmit**(**cfg**->**spi**,** **&**cmd**,** 1**,** 100**);**

// Blokada wejścia zegarowego

HAL\_GPIO\_WritePin**(**cfg**->**ce\_port**,** cfg**->**ce\_pin**,** GPIO\_PIN\_SET**);**

**}**

void display\_write\_data**(**struct display\_config **\*** cfg**,** uint8\_t data**)** **{**

// Wybór tryby transmisji poleceń

HAL\_GPIO\_WritePin**(**cfg**->**dc\_port**,** cfg**->**dc\_pin**,** GPIO\_PIN\_SET**);**

// Odblokowanie wejścia zegarowego

HAL\_GPIO\_WritePin**(**cfg**->**ce\_port**,** cfg**->**ce\_pin**,** GPIO\_PIN\_RESET**);**

// Transmisja danych interfejsem SPI

HAL\_SPI\_Transmit**(**cfg**->**spi**,** **&**data**,** 1**,** 100**);**

// Blokada wejścia zegarowego

HAL\_GPIO\_WritePin**(**cfg**->**ce\_port**,** cfg**->**ce\_pin**,** GPIO\_PIN\_SET**);**

**}**

**Listing 3. Plik programu głównego *main.c***

**[...]**

/\* USER CODE BEGIN Includes \*/

#include "display.h"

/\* USER CODE END Includes \*/

**[...]**

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

**{**

**[...]**

/\* USER CODE BEGIN 2 \*/

struct display\_config cfg**;**

cfg**.**spi **=** **&**hspi1**;**

cfg**.**reset\_port **=** RST\_GPIO\_Port**;**

cfg**.**reset\_pin **=** RST\_Pin**;**

cfg**.**bl\_port **=** BL\_GPIO\_Port**;**

cfg**.**bl\_pin **=** BL\_Pin**;**

cfg**.**dc\_port **=** DC\_GPIO\_Port**;**

cfg**.**dc\_pin **=** DC\_Pin**;**

cfg**.**ce\_port **=** CE\_GPIO\_Port**;**

cfg**.**ce\_pin **=** CE\_Pin**;**

display\_setup**(&**cfg**);**

display\_set\_dxy**(&**cfg**,** horizontal**,** 0**,** 0**);**

**for** **(**uint8\_t row **=** 0**;** row **<** 6**;** row**++)** **{**

**for** **(**uint8\_t col **=** 0**;** col **<** 10**;** col**++)** **{**

**for** **(**uint8\_t j **=** 0**;** j **<** 4**;** j**++)** display\_write\_data**(&**cfg**,** 0x0F**);**

**for** **(**uint8\_t j **=** 0**;** j **<** 4**;** j**++)** display\_write\_data**(&**cfg**,** 0xF0**);**

**}**

**for** **(**uint8\_t j **=** 0**;** j **<** 4**;** j**++)** display\_write\_data**(&**cfg**,** 0x0F**);**

**}**

/\* USER CODE END 2 \*/

**[...]**

**}**

**[...]**

**Listing 4. Dodanie bufora znaków w pliku *display.h***

#ifndef display\_header

#define display\_header

**[...]**

static const uint8\_t display\_font**[][**5**]** **=**

**{**

**{** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00 **},**

**{** 0x00**,** 0x00**,** 0x5f**,** 0x00**,** 0x00 **},**

**{** 0x00**,** 0x07**,** 0x00**,** 0x07**,** 0x00 **},**

**[...]** // Pełny kod dostępny do pobrania z serwera FTP

**{** 0x00**,** 0x41**,** 0x36**,** 0x08**,** 0x00 **},**

**{** 0x10**,** 0x08**,** 0x08**,** 0x10**,** 0x08 **},**

**{** 0x78**,** 0x46**,** 0x41**,** 0x46**,** 0x78 **}**

**};**

// Powyższa czcionka, autorstwa Pascala Stanga, pochodzi

// z biblioteki GLCD, dostępnej na licencji BSD. Źródło:

// https://github.com/andygock/glcd/blob/master/fonts/font5x7.h

**[...]**

void display\_write\_char**(**struct display\_config **\*** cfg**,** char chr**);**

void display\_rewrite\_buffer**(**struct display\_config **\*** cfg**);**

void display\_clear\_buffer**(**struct display\_config **\*** cfg**);**

**[...]**

#endif

**Listing 5. Definicje funkcji wyświetlających znaki w *display.c***

**[...]**

void display\_write\_char**(**struct display\_config **\*** cfg**,** char chr**)**

**{**

**for** **(**uint8\_t i **=** 0**;** i **<** 5**;** i**++)**

display\_write\_data**(**cfg**,** display\_font**[**chr **-** 0x20**][**i**]);**

display\_write\_data**(**cfg**,** 0x00**);**

**}**

void display\_rewrite\_buffer**(**struct display\_config **\*** cfg**)**

**{**

display\_set\_dxy**(**cfg**,** horizontal**,** 0**,** 0**);**

**for(**uint8\_t i**=**0**;** i**<**5**;** i**++)**

**for(**uint8\_t j**=**0**;** j**<**14**;** j**++)**

display\_write\_char**(**cfg**,** cfg**->**buffer**[**i**][**j**]);**

**}**

void display\_clear\_buffer**(**struct display\_config **\*** cfg**)**

**{**

**for(**uint8\_t i**=**0**;** i**<**6**;** i**++)**

**for(**uint8\_t j**=**0**;** j**<**14**;** j**++)**

cfg**->**buffer**[**i**][**j**]** **=** ' '**;**

**}**

**Listing 6. Modyfikacja programu głównego *main.c***

**[...]**

/\* USER CODE BEGIN Includes \*/

#include "string.h"

#include "display.h"

/\* USER CODE END Includes \*/

**[...]**

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

**{**

**[...]**

/\* USER CODE BEGIN 2 \*/

struct display\_config cfg**;**

cfg**.**spi **=** **&**hspi1**;**

cfg**.**reset\_port **=** RST\_GPIO\_Port**;**

cfg**.**reset\_pin **=** RST\_Pin**;**

cfg**.**bl\_port **=** BL\_GPIO\_Port**;**

cfg**.**bl\_pin **=** BL\_Pin**;**

cfg**.**dc\_port **=** DC\_GPIO\_Port**;**

cfg**.**dc\_pin **=** DC\_Pin**;**

cfg**.**ce\_port **=** CE\_GPIO\_Port**;**

cfg**.**ce\_pin **=** CE\_Pin**;**

display\_setup**(&**cfg**);**

display\_clear\_buffer**(&**cfg**);**

char hw**[]** **=** "HELLO WORLD!"**;**

memcpy**(&(**cfg**.**buffer**[**2**][**1**]),** hw**,** strlen**(**hw**));**

display\_rewrite\_buffer**(&**cfg**);**

/\* USER CODE END 2 \*/

**[...]**

**}**

**[...]**