Listing 1. Ramka danych włączająca na kolejnych 30 diodach na pasku naprzemiennie kolory czerwony, zielony i niebieski

@255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255\n\r

Listing 2. Szablon modyfikacji pliku main.c

/\* USER CODE BEGIN Includes \*/

#include „frame\_parser.h”

#include „ws2812b.h”

/\* USER CODE END Includes \*/

/\* USER CODE BEGIN 0 \*/

volatile ws2812b\_config led\_strip\_handler**;**

uint8\_t diodes\_count **=** 30**;**

volatile struct frame\_parser\_state frame\_parser\_handler**;**

volatile uint8\_t recv\_char**;**

void HAL\_UART\_RxCpltCallback**(**UART\_HandleTypeDef **\*** uart**)**

**{**

**if** **(**uart **==** **&**huart1**)**

**{**

frame\_parser\_recv\_char**(&**frame\_parser\_handler**,** recv\_char**);**

HAL\_UART\_Receive\_IT**(&**huart1**,** **&**recv\_char**,** 1**);**

**}**

**}**

void setup\_uart**(**UART\_HandleTypeDef **\*** uart**)**

**{**

HAL\_Delay**(**1000**);**

HAL\_UART\_Transmit**(**uart**,** „AT+BAUD4”**,** strlen**(**„AT+BAUD4”**),** 100**);**

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

HAL\_UART\_Transmit**(**uart**,** „AT+NAMEep\_bt”**,** strlen**(**„AT+NAMEep\_bt”**),** 100**);**

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

HAL\_UART\_Transmit**(**uart**,** „AT+PIN3498”**,** strlen**(**„AT+PIN3498”**),** 100**);**

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

**}**

/\* USER CODE END 0 \*/

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

**{**

/\* USER CODE BEGIN 1 \*/

/\* USER CODE END 1 \*/

/\* MCU Configuration \*/

/\* Reset of all peripherals, Initializes the Flash interface and the Systick. \*/

HAL\_Init**();**

/\* Configure the system clock \*/

SystemClock\_Config**();**

/\* Initialize all configured peripherals \*/

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

MX\_SPI1\_Init**();**

MX\_USART1\_UART\_Init**();**

/\* USER CODE BEGIN 2 \*/

ws2812b\_config led\_strip\_handler **=** ws2812b\_init**(&**hspi1**,** diodes\_count**);**

setup\_uart**(&**huart1**);**

frame\_parser\_handler **=** frame\_parser\_init**(&**led\_strip\_handler**,** diodes\_count**);**

HAL\_UART\_Receive\_IT**(&**huart1**,** **&**recv\_char**,** 1**);**

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

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

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

frame\_parser\_process\_frame**(&**frame\_parser\_handler**);**

**}**

/\* USER CODE END 3 \*/

**}**

**Listing 3. Plik Inc/frame\_parser.h**

#ifndef frame\_parser\_header

#define frame\_parser\_header

#include „stdint.h”

#include „stdlib.h”

#include „string.h”

#include „main.h”

#include „ws2812b.h”

struct frame\_parser\_state

**{**

ws2812b\_config **\*** led\_strip\_handler**;**

uint8\_t diodes\_count**;**

uint8\_t line\_buffer**[**800**];**

uint8\_t writer\_position**;**

uint8\_t reader\_position**;**

uint8\_t field\_buffer**[**30**];**

uint8\_t field\_position**;**

uint8\_t process\_frame**;**

**};**

struct frame\_parser\_state frame\_parser\_init**(**ws2812b\_config **\*** \_led\_strip\_handler**,** uint8\_t \_diodes\_count**);**

void frame\_parser\_recv\_char**(**struct frame\_parser\_state **\*** state**,** uint8\_t recv\_char**);**

void frame\_parser\_read\_field**(**struct frame\_parser\_state **\*** state**);**

void frame\_parser\_process\_frame**(**struct frame\_parser\_state **\*** state**);**

#endif

**Listing 4. Plik Src/frame\_parser.c**

#include „frame\_parser.h”

struct frame\_parser\_state frame\_parser\_init**(**ws2812b\_config **\*** \_led\_strip\_handler**,** uint8\_t \_diodes\_count**)**

**{**

struct frame\_parser\_state state**;**

state**.**led\_strip\_handler **=** \_led\_strip\_handler**;**

state**.**diodes\_count **=** \_diodes\_count**;**

**for(**uint16\_t i**=**0**;** i**<**800**;** i**++)** state**.**line\_buffer**[**i**]** **=** ‚\0’**;**

state**.**writer\_position **=** 0**;**

state**.**reader\_position **=** 0**;**

**for(**uint8\_t i**=**0**;** i**<**30**;** i**++)** state**.**field\_buffer**[**i**]** **=** ‚\0’**;**

state**.**field\_position **=** 0**;**

state**.**process\_frame **=** 0**;**

**return** state**;**

**}**

void frame\_parser\_recv\_char**(**struct frame\_parser\_state **\*** state**,** uint8\_t recv\_char**)**

**{**

**if** **(**state**->**writer\_position **==** 0 **&&** recv\_char **==** ‚@’**)**

**{**

state**->**writer\_position**++;**

**}** **else** **if** **(**state**->**writer\_position **>=** 1 **&&** state**->**writer\_position **<** 799**)**

**{**

**if** **(**recv\_char **==** ‚\r’ **||** recv\_char **==** ‚\n’**)**

**{**

state**->**line\_buffer**[**state**->**writer\_position **-** 1**]** **=** ‚\0’**;**

state**->**writer\_position **=** 0**;**

state**->**process\_frame **=** 1**;**

**}** **else**

**{**

state**->**line\_buffer**[**state**->**writer\_position **-** 1**]** **=** recv\_char**;**

state**->**writer\_position**++;**

**}**

**}** **else**

**{**

state**->**writer\_position **=** 0**;**

**}**

**}**

void frame\_parser\_read\_field**(**struct frame\_parser\_state **\*** state**)**

**{**

state**->**field\_position **=** 0**;**

**while(**state**->**line\_buffer**[**state**->**reader\_position**]** **!=** ‚,’ **&&** state**->**line\_buffer**[**state**->**reader\_position**]** **!=** ‚\0’

**&&** state**->**field\_position **<** 29**)** **{**

state**->**field\_buffer**[**state**->**field\_position**]** **=** state**->**line\_buffer**[**state**->**reader\_position**];**

state**->**reader\_position**++;**

state**->**field\_position**++;**

**}**

state**->**field\_buffer**[**state**->**field\_position**]** **=** ‚\0’**;**

state**->**reader\_position**++;**

**}**

void frame\_parser\_process\_frame**(**struct frame\_parser\_state **\*** state**)**

**{**

**if(**state**->**process\_frame **==** 0**)** **return;**

state**->**process\_frame **=** 0**;**

state**->**reader\_position **=** 0**;**

**for(**int i**=**0**;** i**<**30**;** i**++)**

**{**

ws2812b\_color rgb**;**

frame\_parser\_read\_field**(**state**);**

sscanf**(**state**->**field\_buffer**,** „%d”**,** **&(**rgb**.**red**));**

frame\_parser\_read\_field**(**state**);**

sscanf**(**state**->**field\_buffer**,** „%d”**,** **&(**rgb**.**green**));**

frame\_parser\_read\_field**(**state**);**

sscanf**(**state**->**field\_buffer**,** „%d”**,** **&(**rgb**.**blue**));**

ws2812b\_set\_diode\_color**(**state**->**led\_strip\_handler**,** i**,** rgb**);**

**}**

ws2812b\_refresh**(**state**->**led\_strip\_handler**);**

**}**

**Listing 5. Plik Inc/ws2812b.h**

#ifndef ws2812b\_header

#define ws2812b\_header

#include „stm32f4xx\_hal.h”

#include „spi.h”

**typedef** struct ws2812b\_color

**{**

uint8\_t red**,** green**,** blue**;**

**}** ws2812b\_color**;**

**typedef** struct ws2812b\_config

**{**

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

uint16\_t diodes\_count**;**

ws2812b\_color **\*** colors\_array**;**

**}** ws2812b\_config**;**

ws2812b\_config ws2812b\_init**(**SPI\_HandleTypeDef **\*** spi\_handler**,** uint16\_t diodes\_count**);**

void ws2812b\_set\_diode\_color**(**ws2812b\_config **\*** config**,** uint16\_t diode\_id**,** ws2812b\_color color**);**

void ws2812b\_refresh**(**ws2812b\_config **\*** config**);**

#endif

**Listing 6. Plik Src/ws2812b.c**

#include „ws2812b.h”

ws2812b\_config ws2812b\_init**(**SPI\_HandleTypeDef **\*** spi\_handler**,** uint16\_t diodes\_count**)**

**{**

ws2812b\_config config**;**

config**.**spi\_handler **=** spi\_handler**;**

config**.**diodes\_count **=** diodes\_count**;**

config**.**colors\_array **=** calloc**(**diodes\_count**,** **sizeof(**ws2812b\_color**));**

**return** config**;**

**}**

void ws2812b\_set\_diode\_color**(**ws2812b\_config **\*** config**,** uint16\_t diode\_id**,** ws2812b\_color color**)**

**{**

config**->**colors\_array**[**diode\_id**]** **=** color**;**

**}**

void ws2812b\_refresh**(**ws2812b\_config **\*** config**)**

**{**

const uint8\_t zero **=** 0b00011111**;**

const uint8\_t one **=** 0b00000111**;**

uint8\_t buffer**[**30 **\*** 24**];**

**for** **(**uint16\_t i **=** 0**,** j **=** 0**;** i **<** config**->**diodes\_count**;** i**++)**

**{**

// Green

**for** **(**int16\_t k **=** 7**;** k **>=** 0**;** k**--)**

**{**

**if** **((**config**->**colors\_array**[**i**].**green **&** **(**1 **<<** k**))** **==** 0**)** buffer**[**j**]** **=** one**;**

**else** buffer**[**j**]** **=** zero**;**

j**++;**

**}**

// Red

**for** **(**int16\_t k **=** 7**;** k **>=** 0**;** k**--)**

**{**

**if** **((**config**->**colors\_array**[**i**].**red **&** **(**1 **<<** k**))** **==** 0**)** buffer**[**j**]** **=** one**;**

**else** buffer**[**j**]** **=** zero**;**

j**++;**

**}**

// Blue

**for** **(**int16\_t k **=** 7**;** k **>=** 0**;** k**--)**

**{**

**if** **((**config**->**colors\_array**[**i**].**blue **&** **(**1 **<<** k**))** **==** 0**)** buffer**[**j**]** **=** one**;**

**else** buffer**[**j**]** **=** zero**;**

j**++;**

**}**

**}**

HAL\_SPI\_Transmit**(**config**->**spi\_handler**,** **&**buffer**,** config**->**diodes\_count **\*** 24**,** 1000**);**

HAL\_Delay**(**1**);**

**}**

Listing 7. Testowa ramka danych

@255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255,255,0,0,0,255,0,0,0,255