

Teaching online electronics, microcontrollers and programming in Higher Education

### **Programing of embedded systems**

6. Neopixels driver

Lead Partner: Warsaw University of Technology

**Authors: Daniel Krol** 

University of Applied Sciences in Tarnow



6. Neopixels driver

## Declaration

This report has been prepared in the context of the ENGINE project. Where other published and unpublished source materials have been used, these have been acknowledged.

## Copyright

#### © Copyright 2021 - 2023 the ENGINE Consortium

Warsaw University of Technology (Poland)

International Hellenic University (IHU) (Greece)

European Lab for Educational Technology- EDUMOTIVA (Greece)

University of Padova (Italy)

University of Applied Sciences in Tarnow (Poland)

All rights reserved.



This document is licensed to the public under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

## **Funding Disclaimer**

This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

### 6. Neopixels driver

#### I. Wykorzystanie interfejsu SPI

- 1. Stwórz nowy projekt dla płyty *LPCXpresso804*, tak jak na poprzednich zajęciach i nazwij go *Lab06*.
- 2. Dodaj sterownik interfejsu SPI:

ect name: ZPSW_Lab05		Project name suffix: Search			
Use default location					
evice Packages	Board	Project Type	Project Options		
LPC804M101JDH24 LPC804M101JDH20 LPC804M111JDH20	Default board files     Empty board files	C Project C++ Project C Static Library C++ Static Library	SDK Debug Console Semihost CMSIS-Core Copy sources Import other files	O UART	
omponents			Components selection summary	E	
dd or remove SDK software compone	its				
Drivers type to filter Name KB iap	Description IAP Driver	Version Info	Drivers     Operating Systems     Utilities		
🗹 🗛 iocon	IOCON Driver LPC_ACOMP Driver CBC Driver	2.0.1 2.1.0 2.1.1			
Age (pc_acomp     Age (pc_crc     Age mrt     Age pint	MRT Driver PINT Driver	2.0.4			
	MRT Driver PINT Driver PLU Driver Power Driver Reset Driver Rem_api Driver	2.04 2.19 2.2.1 2.0.0 2.1.2 2.0.1			
Alipe_acomp     Alipe_acomp     Alipe_acomp     Alipe_crc     Amot     Amot	MRT Driver PIND Driver PLUD Driver Power Driver Reset Driver Rom_api Driver SPI Driver SWM Driver Swm_connections Driver SYSCON Driver	2.0.4 2.19 2.21 2.00 2.12 2.0.1 2.0.4 2.0.2 2.0.1 2.0.1 2.0.1			

3. Przejdź do Config Tool ->Pins a następnie w Functional group stwórz nowy preset BOARD\_InitNeopixelsPin:

• • •		Functional group properties
Functional groups 🕕 🗋 🚷 🔨 🗸	Name:	BOARD_InitNeopixeIsPin
BOARD InitPine		Called by default initialization function
BOARD Init! EDsPins		Set custom #define prefix
BOARD_InitDEBUG_UARTPins	Prefix:	BOARD INITNEOPIXELSPIN
BOARD_InitSWD_DEBUGPins		
BOARD_Initl2CPins		
BOARD_InitBUTTONsPins		Full pins initialization
BOARD_InitNeopixeIsPin		De-initialization function
		Cancel OK

4. Podłącz linię *MOSI* interfejsu *SPI0* do wyprowadzenia *PIO\_010* mikrokontrolera:

000	Image: State of the state of																					
1 11 - 10 K	2PSW_Lat	05	🔻 🏤 🖪	Update Code • Fur	ctional Group	BOARD_InitN	leopixelsP	in 💌 🏓 🗐	. 4 🗞 (		1.2.5	81 🕸 - 💁	- 0 /	<b>9 • 1</b> 1 2 • •	8 - <b>10 47</b> 4	• • •				Q 🔡	K 🖷 🗤 🕴 🔿	8
Pins 🕄	🐼 Peripheral Signa	ls						😳 Package 😫 🧔 Ex	pansion Hea	der					ର୍ ପ ଦ	0 🖃 = 🗆	A Overview 😫	Code P	review 🔠 Registe	ers		
990	₩₩ -000	🗲 🕇 😥 🔍 typ															> Configura	ation - Gene	ral Info			
C      C	Comparing a second	05	A Constraint of the second secon	Notate Code • Fur Andore UHO R3 Co cr42(3) (20) cr42(3)	spi spi spi spi spi spi spi spi spi spi	BOMD HINK     BOMD HINK	C WINCHER	Periphe Peripheriphe Periphe Periphe Periphe Periphe Periphe Periphe	ACMP Reparation Heat	ADC0 O	anno tils view.			Comparing the second seco			Councileur 25 Councileur 25 Councileur Coun	Cocke P Cocke The Cocke	vovier	Q. B. and	electrical din proceeding and another	
Routing D	etails for BOARD Initi	Neo 1 C														1	Problems S				DI	4
a	Peripheral	ional Arrow	Routed pin/sig I ahe	l Identifier	Direction	GPIO initia	il sta Morie	a Invert	Hysteresis	Open drai	n DAC M	ide					type niter text	Decent		leas		
11	SPIO S	AOSI	[11] PIDO_10 CN6]	(11]/VR1/C n/a	Not Spec	ified n/a	PullU	Disabled	Enabled	Disabled	n/a						Seven Sarning	USARTO I2C0 SPI0		Issue Periphe Periphe	al USARTO is not initializ al I2CO is not initializ al SPIO is not initializ	sali red zed
ZPSW_Lab0	5																		U NXP	LPC804* (ZPSW.	Lab05)	-

5. W zakładce *Clocks* kliknij na blok oscylatora *FRO\_OSC* i zmień jego częstotliwość na 24 MHz:



6. Następnie doprowadź sygnał zegara 12 MHz do interfejsu *SPIO* przez wybór *main\_clk* w bloku *SPIOCLKSEL*:

• • •	e workspace_LPC55s69 - ZPSW_Lab05	5/source/ZPSW_Lab05.c - MCUXpresso IDE	
• 🔛 🐚 📸 ZPSW_Lab05	🔻 👫 🛕 📓 Update Code 🔹 Functional Group 🛛 🛛 🗛 RD_BootClockFR018M 🔍 🏲 💭 🤣 😓 🛛	s 🗈 🖷 3. The set 🕼 💊 🖌 🖌 🍘 🖋 🖉 🖉 🖓 🖓 🖉 🖉 🖉 🖉 🖉 🖉	Q 📾 🗶 📾 I 🗘
Clocks Table 🛟 Clocks Diagram 💱		→ Q Q P I = SYSCON.SPIOCLKSEL •	🗆 🔺 Overview 🚡 Code Previ 🔢 Registers 🧮 Details 😫 🔕 Clock Cons 📟
un Mode ACTIVE V			T 🖉 🔒 🗛
1		1	Name C., L., Value
			SPI0 clock select main clock
	h annun		
	fro 12 MHz ADCCLKDI	W Jon Jon ADC clock	
1	Disable	d Inactive	
	<i>r</i>		
	fro		
fro FRGOCLKSEL	FROD_MUL FROD_DIV main_clk Inactive	UARTCLK0 clock	
main_cik	+ I I I I I I I I I I I I I I I I I I I	Inactive	
V	1256 /256 → frg0_c/k		
	fro UANI ICLKBEL		
	frg0.ck	ULARTCLK1 clock	
	fro_div	- Incore	
	fro		
	main_clk inactive	- IZCELKO cinek	
	frg0_ck	PL-P Inactive	
	fro		
	main_cit inactive	120CLK1 clock	
	fro_div	- nactive	
	<b>b</b>		
	fro SHOCLKSEL		
	frg0_ck 12 MHz	SPICLK0 clock	🊵 Problems 🕄 🛛 🖪 🍸 🐡
	fro_div		
	· · · · · · · · · · · · · · · · · · ·		Level v Resource Issue
divto750k 750 kHz		NTL-N FBO clock do to 250k	
		750 kHz	
	lposc_clk	D-D LowPower clock	
		1 MHz	
		-> WWDT clock	
i		1 MH2	
< 1 xh05			i dha anna anna a ranna a san
			: O NXP LPC804* (ZPSW_Labos)

7. W zakładce *Peripherals* wybierz sterownik *SPI*:

000				o workspace_LPC55s69 - ZP	SW_Lab05/source/ZPSW_Lab0	15.c - MCUXpresso IDE		
📑 • 📄 🌇 👔 ZPSW_Lab05	💌 🌴 🔺	Update Code + Function	nal Group BOARD_I	nitPeripherals 💌 🌹 🗐 🞺	9 💼 N 💷 19 🖄 🖉 🖉	8 .R 6+ 💁 🔗 - 刘 :	원 - 원 - 🏷 🗘 🗇 - 아 - 📷	Q 🔡 🗶 🖷 🗤 🕈 🗘 🛗
Components 🔉 🦞 Peripherals		overview 😫 🔂 Code	Preview					
type filter text		> Configuration - Ger	neral Info					
	<u> </u>	<ul> <li>Configuration - HW</li> </ul>	/ Info					
Peripheral drivers (Device specific)	0	Processor: LPC80	4					
		Part number: LPC80	4M101JDH24					
Custom initialization	0	Core: Cortex	-MOP					
		Board: LPCXp	resso804					
		SDK Version: ksdk2_	_0					
		> Project						
		<ul> <li>Peripherals</li> </ul>						
			ofigures the initializ	Sele	ct configuration component			
		(1)	-					
			5	elect which components should be offered	Present in the tool-chain project V			
		<ul> <li>Generated code</li> </ul>	Co	infiguration component Component descript	ion	Category		
		✓ Update code enable	ed 🗕	GPIO General Purpose I/O	(GPIO)	Peripheral drivers (Device s		
		board/peripherals	LC -	USART Universal Synchrono	us/Asynchronous Receiver/Transmitte	r (USART) Peripheral drivers (Device s		
		board/peripherals	.h					
		<ul> <li>Functional groups</li> </ul>						
		BOARD_InitPeript	berals 🌇					
		<ul> <li>Other tools</li> </ul>						
			$\sim$	Open SDK components manager	Cancel	ОК		
			(יווי)					
		<b>()</b>						
		2 Problems						в 🍸 😑 🗆
		Level v Resource	e	Issue	Origin	Target	Туре	
		& Warning USARTO	)	Peripheral USART0 is not initialized	Pins:BOARD_InitDEBUG_UARTPins	Peripherals: BOARD_InitPeripherals	Validation	
		Warning 12C0		Peripheral I2C0 is not initialized Peripheral SPID is not initialized	Pins:BOARD_InitI2CPins Pins:BOARD_InitI2CPins	Peripherals: BOARD_InitPeripherals Peripherals: BOARD_InitPeripherals	Validation	
		- manalig delu		r engeneral en te la fret initialized	- master ma_matterprotection			
ZPSW_Lab05								O NXP LPC804* (ZPSW. Lab05)

8. Skonfiguruj parametry transmisji SPI w trybie Polling dla interfejsu SPIO:



9. Przejdź do głównego pliku projektu i zmodyfikuj kod jak poniżej:

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LCR804.h"
#include "fsl_debug_console.h"
#define LEDS 37
#define GET_BIT(k, n)
#define SET_BIT(k, n)
#define CLR_BIT(k, n)
                                      #define CODE_0
#define CODE_1
                                                                           0b10000
                                                                           0b11100
uint32_t colors[LEDS]={0};
for(int j=0;j<n;j++) {
    for(int i=23;i>=0;i--) {
        LED_data = GET_BIT(value[j], i) ? CODE_1 : CODE_0;
        COV_CTAT_TXRDY_MASK));

                                            while(!(base->STAT & SPI_STAT_TXRDY_MASK));
base->TXDAT = LED_data ;
                              }
               }
                // Reset >= 50 us
               LED_data=0;
               for(int j=0;j<50;j++) {
    while(!(base->STAT & SPI_STAT_TXRDY_MASK));
    base->TXDAT = LED_data ;
               }
3
  * @brief Application entry point.
int main(void) {
                   Init board hardware. */
               BOARD_InitBootPins();
BOARD_InitBootClocks();
BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
                                debug console.
               BOARD_InitDebugConsole();
#endif
```



10. Podłącz moduł LED do płytki prototypowej według poniższego schematu:



11. Zbuduj projekt w konfiguracji **Release**, zaprogramuj układ i sprawdź działanie programu.

**UWAGA!** Nie wolno włączać **wszystkich diod z pełną jasnością** bez dodatkowego zasilania modułu - grozi uszkodzeniem stabilizatora. Płyta prototypowa jest w stanie zasilić cały moduł LED z ok. **1/16 jasności** maksymalna wartość dla koloru białego: HEX **0x0F0F0F**. DEC R:**15**, G:**15**, B:**15** 

6. Neopixels driver

#### II. Funkcja setRGB

- 1. Napisz funkcję ustawiająca kolor w formacie RGB 888
- 2. Dodaj timer systemowy i sprawdź regulację jasności poszczególnych składowych RGB:

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC804.h"
#include "fsl_debug_console.h"
#define LEDS 37
#define GET_BIT(k, n)
#define SET_BIT(k, n)
#define CLR_BIT(k, n)
                               (k & (1 << (n)))
(k |= (1 << (n)))
(k &= ~(1 << (n)))
#define CODE_0
#define CODE_1
                                                                     0b10000
                                                                    0b11100
uint32_t colors[LEDS]={0};
uint8_t k=0;
 void Neopixels_Send(SPI_Type *base, uint32_t n, uint32_t *value) {
             uint16_t LED_data=0;
             while(!(base->STAT & SPI_STAT_TXRDY_MASK));
base->TXDAT = LED_data ;
                          }
             }
// Reset >= 50 us
              LED_data=0;
              for(int j=0;j<50;j++) {
    while(!(base->STAT & SPI_STAT_TXRDY_MASK));
    base->TXDAT = LED_data ;
              }
inline uint32_t setRGB(uint8_t r, uint8_t g, uint8_t b) {
             return ((g<<16) | (r<<8) | b);</pre>
ŀ
void SysTick_Handler(void) {
             colors[18] = setRGB(k++, 0,
                                                     0):
             //colors[18] = setRGB(0, k++, 0);
//colors[18] = setRGB(0, 0, k++);
Neopixels_Send(SPI0_PERIPHERAL, LEDS, colors);
}
 * @brief Application entry point.
int main(void) {
               /* Init board hardware. */
              BOARD_InitBootPins();
             BOARD_InitBootClocks();
BOARD_InitBootPeripherals()
 #ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
                           SL debug conso
                                               le.
             BOARD_InitDebugConsole();
 #endif
             SPI_WriteConfigFlags(SPI0_PERIPHERAL, kSPI_ReceiveIgnore);
             SysTick_Config(SystemCoreClock / 100); // 100 Hz
             while(1) {
              return 0 :
}
```

3. Zbuduj projekt, zaprogramuj układ i sprawdź działanie programu.

6. Neopixels driver

#### II. Funkcja setBrightness

1. Napisz funkcję ustawiająca jasność zadanego koloru:

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC804.h"
#include "fsl_debug_console.h"
 #define LEDS 37
#define GET_BIT(k, n)
#define SET_BIT(k, n)
#define CLR_BIT(k, n)
                                   #define CODE_0
#define CODE_1
                                                                       0b10000
                                                                       0b11100
uint32_t colors[LEDS]={0};
uint8_t k=0;
 void Neopixels_Send(SPI_Type *base, uint32_t n, uint32_t *value) {
              uint16_t LED_data=0;
              for(int j=0;j<n;j++) {
    for(int i=23;i>=0;i--) {
                                          LED_data = GET_BIT(value[j], i) ? CODE_1 : CODE_0;
                                           while(!(base->STAT & SPI_STAT_TXRDY_MASK));
                                          base->TXDAT = LED_data ;
                            }
              }
               // Reset >= 50 us
              LED data=0:
              LED_uata=v,
for(int j=0;j<50;j++) {
    while(!(base->STAT & SPI_STAT_TXRDY_MASK));
    base->TXDAT = LED_data ;
              }
}
inline uint32_t setRGB(uint8_t r, uint8_t g, uint8_t b) {
              return ((g<<16) | (r<<8) | b);</pre>
}
uint32_t setBrightness(uint32_t color, uint8_t level) {
              uint8_t b = level * (color & 0x0000FF) / 255;
uint8_t r = level * ((color & 0x00FF00) >> 8) / 255;
uint8_t g = level * ((color & 0xFF0000) >> 16) / 255;
              return ((g<<16) | (r<<8) | b);</pre>
ì
void SysTick_Handler(void) {
              colors[18] = setRGB(255, 0, 128);
colors[18] = setBrightness(colors[18], k++);
              Neopixels_Send(SPI0_PERIPHERAL, LEDS, colors);
1
  * @brief Application entry point.
 int main(void) {
                 Init board hardware. */
              BOARD_InitBootPins();
BOARD_InitBootClocks();
BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
                               de
              BOARD_InitDebugConsole();
#endif
              SPI_WriteConfigFlags(SPI0_PERIPHERAL, kSPI_ReceiveIgnore);
              SysTick_Config(SystemCoreClock / 100); // 100 Hz
              while(1) {
              return 0 ;
3
```

2. Zbuduj projekt, zaprogramuj układ i sprawdź działanie programu dla różnych kolorów RGB.

6. Neopixels driver

#### III. Prosta animacja

```
1. Napisz funkcję "przesuwającą" świecący punkt po matrycy:
```

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC804.h"
#include "fsl_debug_console.h"
 #define LEDS 37
#define GET_BIT(k, n)
#define SET_BIT(k, n)
#define CLR_BIT(k, n)
                                 #define CODE_0
#define CODE_1
                                                               0b10000
                                                               0b11100
uint32_t colors[LEDS]={0};
uint8_t k=0;
uint32_t rgbColor=0;
 void Neopixels_Send(SPI_Type *base, uint32_t n, uint32_t *value) {
            uint16_t LED_data=0;
             for(int j=0;j<n;j++) {
    for(int i=23;i>=0;i--) {
                                      LED_data = GET_BIT(value[j], i) ? CODE_1 : CODE_0;
                                      while(!(base->STAT & SPI_STAT_TXRDY_MASK));
                                      base->TXDAT = LED_data ;
                         }
            }
              // Reset >= 50 us
             LED data=0;
             LED_uata=v,
for(int j=0;j<50;j++) {
    while(!(base->STAT & SPI_STAT_TXRDY_MASK));
    base->TXDAT = LED_data ;
             }
}
inline uint32_t setRGB(uint8_t r, uint8_t g, uint8_t b) {
             return ((g<<16) | (r<<8) | b);</pre>
3
 void Animate(uint32_t color)
             colors[k]=color;
             if(k>=LEDS)
                          k=0:
}
void SysTick_Handler(void) {
             rgbColor = setRGB(0, 0, 15); // MAX RGB: (15, 15, 15)
             Animate(rgbColor);
            Neopixels_Send(SPI0_PERIPHERAL, LEDS, colors);
}
 * @brief Application entry point.
int main(void) {
              /* Thit board hardware. */
             BOARD_InitBootPins();
             BOARD_InitBootClocks();
BOARD_InitBootPeripherals()
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
                           debug conso
                                           .е.
             BOARD_InitDebugConsole();
 #endif
            SPI_WriteConfigFlags(SPI0_PERIPHERAL, kSPI_ReceiveIgnore);
             SysTick_Config(SystemCoreClock / 10); // 10 Hz
             while(1) {
             return 0 :
}
```

2. Zbuduj projekt, zaprogramuj układ i sprawdź działanie programu.

6. Neopixels driver

### IV. Animacja LUT (lookup table)

1. Napisz animację wykorzystującą tablicę LUT. Nie przekraczaj wartości 15 na poszczególnych składowych RGB!

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC804.h"
#include "fsl_debug_console.h"
#define LEDS 37
#define GET_BIT(k, n)
#define SET_BIT(k, n)
#define CLR_BIT(k, n)
                                     #define CODE_0
                                                                      0b10000
#define CODE_1
                                                                      0b11100
uint32_t colors[LEDS]={0};
uint8_t k=0;
uint32_t rgbColor=0;
const bool pic1[4][LEDS] = {{
                               0,0,0,0,
0,0,0,0,0,0,
                              0,0,0,0,0,0,0,
                            0,0,0,1,0,0,0,
0,0,0,0,0,0,0,
                               0.0.0.0.0.
                                0,0,0,0,
},{
                                0,0,0,0,
                              0,0,0,0,0,
0,0,1,1,0,0,
                            0,0,1,0,1,0,0,
0,0,1,1,0,0,
                               0,0,0,0,0,0,
                                0,0,0,0,
},{
                              0,0,0,0,
0,1,1,1,0,
                              0,1,0,0,1,0,
                            0,1,0,0,0,1,0,
0,1,0,0,1,0,
                               0,1,1,1,0,
0,0,0,0,
},{
                                1,1,1,1,
                               1.0.0.0.1
                              1,0,0,0,0,1,
                            1.0.0.0.0.0.1
                              1,0,0,0,0,1,
                               1.0.0.0.1.
                                1,1,1,1,
}}:
void Neopixels_Send(SPI_Type *base, uint32_t n, uint32_t *value) {
              uint16_t LED_data=0;
              for(int j=0;j<n;j++) {</pre>
                            for(int i=23;i>=0;i--) {
    LED_data = GET_BIT(value[j], i) ? CODE_1 : CODE_0;
                                          while(!(base->STAT & SPI_STAT_TXRDY_MASK));
                                          base->TXDAT = LED_data ;
                            }
              }
// Reset >= 50 us
              LED_data=0;
              for(int j=0;j<50;j++) {
    while(!(base->STAT & SPI_STAT_TXRDY_MASK));
    base->TXDAT = LED_data;
              }
}
inline uint32_t setRGB(uint8_t r, uint8_t g, uint8_t b) {
              return ((q<<16) | (r<<8) | b):
1
void setImage(const bool *image, uint32_t color) {
              for(int i=0; i<LEDS; i++) {</pre>
                            colors[i] = image[i] * color;
              }
lı
void SysTick_Handler(void) {
```

6. Neopixels driver

```
rgbColor = setRGB(0, 0, 15); // MAX RGB: (15, 15, 15)
setImage(pic1[k++ % 4], rgbColor);
            Neopixels_Send(SPI0_PERIPHERAL, LEDS, colors);
}
/*
* @brief Application entry point.
int main(void) {
            /* Init board hardware. */
BOARD_InitBootPins();
BOARD_InitBootClocks();
BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
                            debug conso
                                            e.
            BOARD_InitDebugConsole();
#endif
            SPI_WriteConfigFlags(SPI0_PERIPHERAL, kSPI_ReceiveIgnore);
            SysTick_Config(SystemCoreClock / 5); // 5 Hz
            while(1) {
            return 0 :
3
```

2. Zbuduj projekt, zaprogramuj układ i sprawdź działanie programu.

3. Dodaj kolejną tablicę z animacją:

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC804.h"
#include "fsl_debug_console.h"
#define LEDS 37
#define GET_BIT(k, n)
#define SET_BIT(k, n)
#define CLR_BIT(k, n)
                                                (k & (1 << (n)))
                                            (k \mid = (1 << (n)))
(k \& = ~(1 << (n)))
#define CODE_0
#define CODE_1
                                                                                           0b10000
                                                                                           0b11100
uint32_t colors[LEDS]={0};
uint8 t k=0;
 uint32_t rgbColor=0;
const bool pic1[4][LEDS] = {{
                                         0,0,0,0,
                                      0,0,0,0,0,0,
0,0,0,0,0,0,0,
                                     0,0,0,1,0,0,0,
                                       0.0.0.0.0.0.0.
                                         0,0,0,0,0,
                                          0,0,0,0,
},{
                                          0,0,0,0,
                                     0,0,0,0,0,0,
0,0,1,1,0,0,
0,0,1,0,1,0,0,
                                       0,0,1,1,0,0,
                                        0,0,0,0,0,0,
0,0,0,0,0,
},{
                                          0,0,0,0,
                                    0,0,0,0,0,
0,1,1,1,0,
0,1,0,0,1,0,
0,1,0,0,0,1,0,
0,1,0,0,1,0,
                                        0,1,1,1,0,
0,0,0,0,
},{
                                         1,1,1,1,
                                      1,0,0,0,1,
1,0,0,0,0,1,
                                     1,0,0,0,0,0,1,
1,0,0,0,0,1,
                                        1,0,0,0,1,
1,1,1,1,
};
 const bool pic2[6][LEDS] = {{
                                         1.1.1.1.
                                        0,1,0,1,0,
                                     0,0,1,1,0,0,
0,0,0,1,0,0,0,
                                      0,0,0,0,0,0,0,
0,0,0,0,0,0,
                                          0.0.0.0.
```

### 6. Neopixels driver

```
},{
                              0,0,0,1,
                           0,0,0,1,1,
0,0,0,1,0,1,
                          0,0,0,1,1,1,1,1,
0,0,0,0,0,0,0,0,
                             0,0,0,0,0,0,
                              0,0,0,0,
},{
                              0,0,0,0,
                            0,0,0,0,0,0,
0,0,0,0,0,0,0,
                          0,0,0,1,1,1,1,1,
0,0,0,1,0,1,
                             0,0,0,1,1,
0,0,0,1,
},{
                             0,0,0,0,
0,0,0,0,0,0,
                            0,0,0,0,0,0,0,
                          },{
                              0,0,0,0,
                          1,1,0,0,0,
1,0,0,0,
},{
                             1,0,0,0,
1,1,0,0,0,
                          1,0,1,0,0,0,
1,1,1,1,0,0,0,0
                            0,0,0,0,0,0,0,
                             0,0,0,0,0,0,
0,0,0,0,
}}:
void Neopixels_Send(SPI_Type *base, uint32_t n, uint32_t *value) {
             uint16_t LED_data=0;
             for(int j=0;j<n;j++) {</pre>
                          for(int i=23;i>=0;i--) {
    LED_data = GET_BIT(value[j], i) ? CODE_1 : CODE_0;
                                       while(!(base->STAT & SPI_STAT_TXRDY_MASK));
                                       base->TXDAT = LED_data ;
                          }
             }
              // Reset >= 50 us
             LED_data=0;
             for(int j=0;j<50;j++) {</pre>
                          while(!(base->STAT & SPI_STAT_TXRDY_MASK));
base->TXDAT = LED_data ;
             }
inline uint32_t setRGB(uint8_t r, uint8_t g, uint8_t b) {
             return ((g<<16) | (r<<8) | b);</pre>
lì
void setImage(const bool *image, uint32_t color) {
             for(int i=0; i<LEDS; i++) {</pre>
                          colors[i] = image[i] * color;
             }
void SysTick_Handler(void) {
             rgbColor = setRGB(0, 0, 15); // MAX RGB: (15, 15, 15)
setImage(pic2[k++ % 6], rgbColor);
Neopixels_Send(SPI0_PERIPHERAL, LEDS, colors);
3
 * @brief Application entry point.
int main(void) {
             /* Init board hardware. */
BOARD_InitBootPins();
BOARD_InitBootClocks();
BOARD_InitBootPeripherals()
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
                             debug conso
                                             e.
             BOARD_InitDebugConsole();
#endif
             SPI_WriteConfigFlags(SPI0_PERIPHERAL, kSPI_ReceiveIgnore);
```

```
SysTick_Config(SystemCoreClock / 5); // 5 Hz
while(1) {
}
return 0;
```

4. Zbuduj projekt, zaprogramuj układ i sprawdź działanie programu.

### V. Zadania

}

- 1. Stwórz swoje własne animacje LUT
- 2. Stwórz animację kolorową (pamiętaj żeby nie przekraczać wartości RGB: 15, 15, 15 na wielu diodach)