

Jiří Spáčil

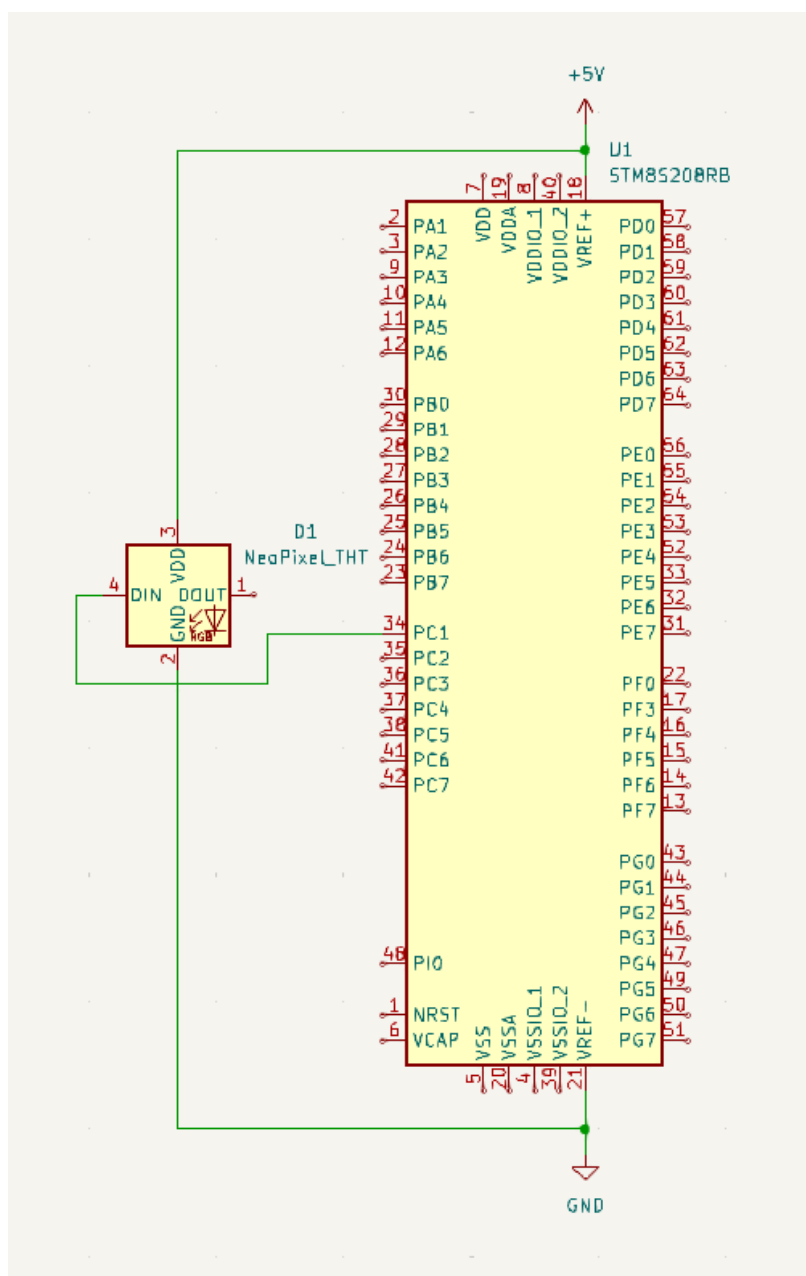
LED efekt – neopixel pásek

samostatný projekt MIT

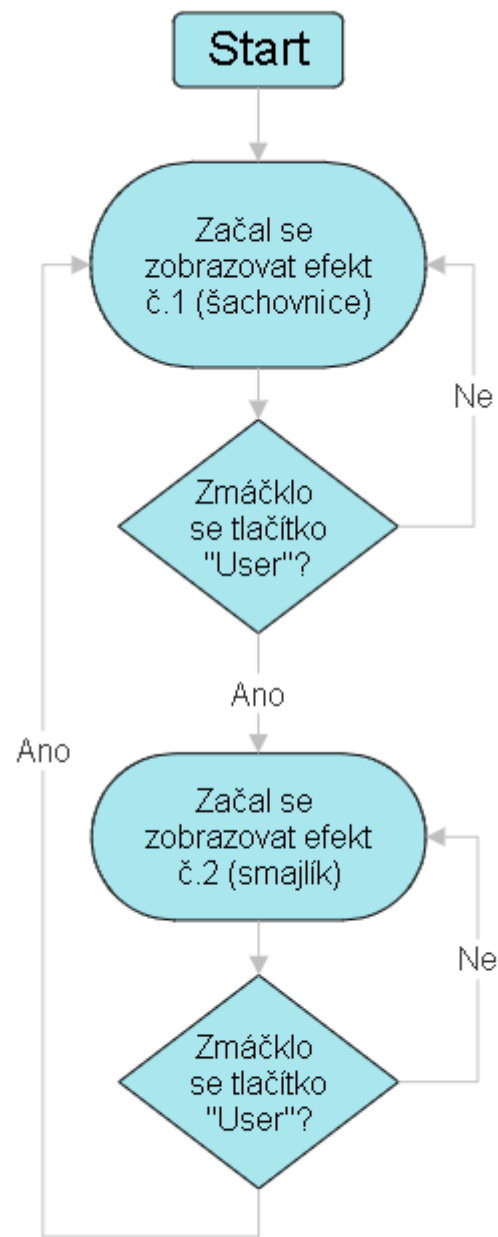
Zadání:

- Po startu se spustí led efekt
- Po stlačení tlačítka se led efekt změní na jiný

Schéma zapojení:



Vývojový diagram:



Popis činnosti:

- Po spuštění se spustí první efekt ve smyčce a bude se opakovat tak dlouho dokud se nezmáčkne tlačítko „User“
- Jakmile se zmáčkne tlačítko „User“ následně se led efekt přepne na jiný, který je také ve smyčce
- Tlačítko funguje jako přepínač mezi led efekty

Zdrojový kód:

```
#include <main.h>
#include <stm8s.h>
#include <neopixel.h> // nadefinování barev, posílání a zobrazování dat
#include <animations.h> // veškeré efekty jsou v této knihovně

#define BUTTON_PORT GPIOE
#define BUTTON_PIN GPIO_PIN_4
#define NEO_WIDTH 8 //nastavení rozsahu zobrazovacího " led pole"
#define NEO_HEIGHT 8 //nastavení rozsahu zobrazovacího " led pole"

uint32_t* color_buffer; //buffer na displej

void setup(void)
{
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); // Set CLK

    GPIO_Init(BUTTON_PORT, BUTTON_PIN, GPIO_MODE_IN_PU_NO_IT);

    init_time();
    init_uart1();
    init_tim();
}
//zkušební kod:
void create_array_one(uint8_t pos){ // dvě ledky se pousunují o ledku dál po 8x8
neopixel poli, stará funkce
    uint8_t pos2 = pos >= 63 ? 0 : pos + 1;
    uint32_t color = _merge(200, 0, 255); //definice barvy ledek
    for(uint8_t i = 0; i < 65; i++){ //posunutí ledek o jednu led
        color_buffer[i] = i == pos || i == pos2 ? color : 0;
    }
    // dvě ledky se pousunují o ledku dál po 8x8 neopixel poli
}

void plot_pixel(uint8_t x, uint8_t y, uint32_t pixel){ //stará funkce
    color_buffer[y*NEO_HEIGHT+x] = pixel;
}

void clear_color_buffer(){ //maže buffer displeje
    for(uint8_t i = 0; i < 65; i++) color_buffer[i] = 0;
}

uint8_t animation_number = 0;
uint32_t (*active_animation)[65]; //pointer na probíhající animaci(sekvenci)
uint8_t active_animation_length = 0; //délka efektu
uint8_t active_frame = 0;
void next_frame(){ //nadefinování pokračování efektu
    active_frame = active_frame + 1;
```

```

    if(active_frame >= active_animation_length) active_frame = 0; // aby se nepřekročila délka efektu
    color_buffer = active_animation[active_frame];
}

int main(void)
{
    setup();
    fill_with_color(200, 0, 255, 64); //definování barvy při resetu/nastavení se rozsvítí celé pole neopixelu
    uint32_t timestamp = 0; // inicializace času
    uint8_t led_pos = 0; //pozice ledky
    uint8_t old_btn_state = 0; //paměť pozice tlačítka
    active_animation_length = 2; //delka naprogramovaného efektu
    active_animation = checks_animation; //kontrola aktivního efektu
    while(1){
        if(GPIO_ReadInputPin(BUTTON_PORT, BUTTON_PIN) == 0){ // inicializace tlačítka
            if(old_btn_state == 0){ // overení zda-li bylo tlačítko zmačknuto
                animation_number++;
                active_frame = 0;
                if(animation_number > 1) animation_number = 0; // detekce zmačknutí tlačítka a následné přidělení efektu
                if(animation_number == 0){
                    active_animation = checks_animation; // zmena z jednoho efektu na druhý
                    active_animation_length = 2; // delka efektu
                }
                else{ // overení zda-li bylo tlačítko zmačknuto
                    active_animation = smile_animation; // zmena z jednoho efektu na druhý
                    active_animation_length = 3; // delka efektu
                }
                old_btn_state = 1;
            }
            else old_btn_state = 0;

            if(milis() - timestamp >= 1000){ // projekce led efektu
                timestamp = milis();
                //led_pos = led_pos >= 63 && direction == 1 ? 0 : led_pos <= 0 && direction == -1 ? 63 : led_pos + direction;
                //clear_color_buffer(); //vyčistí se pole <--- zkušební kod
                next_frame();
                flush(color_buffer); //signál se zobrazí na neopixelu
            }
        }
        return 0;
    }
}

```