

Josef Křivánek

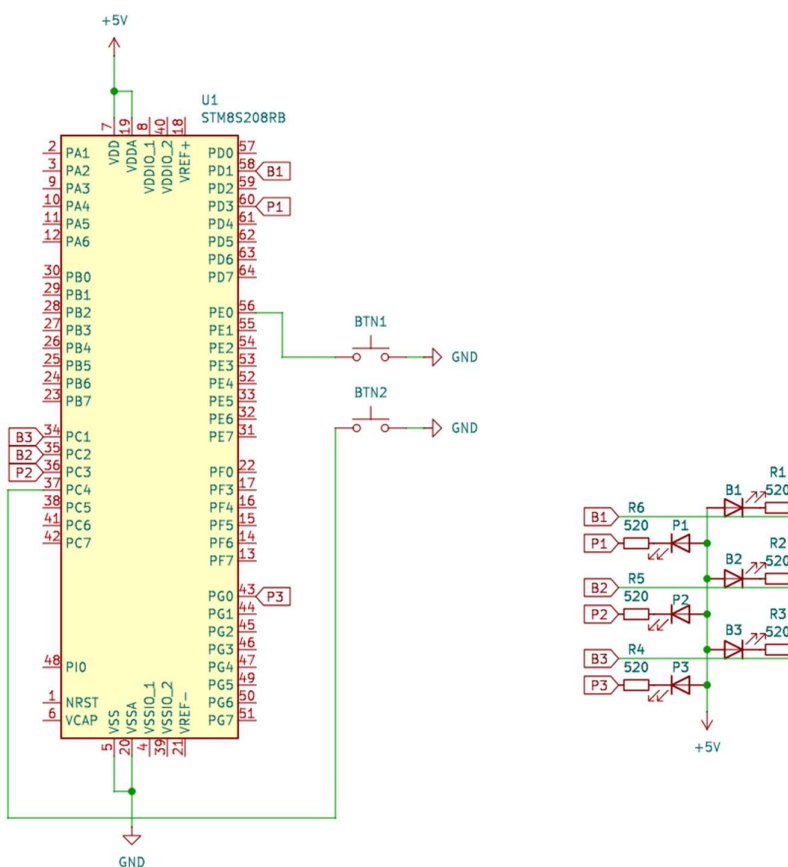
Klikací hra

Samostatný projekt MIT

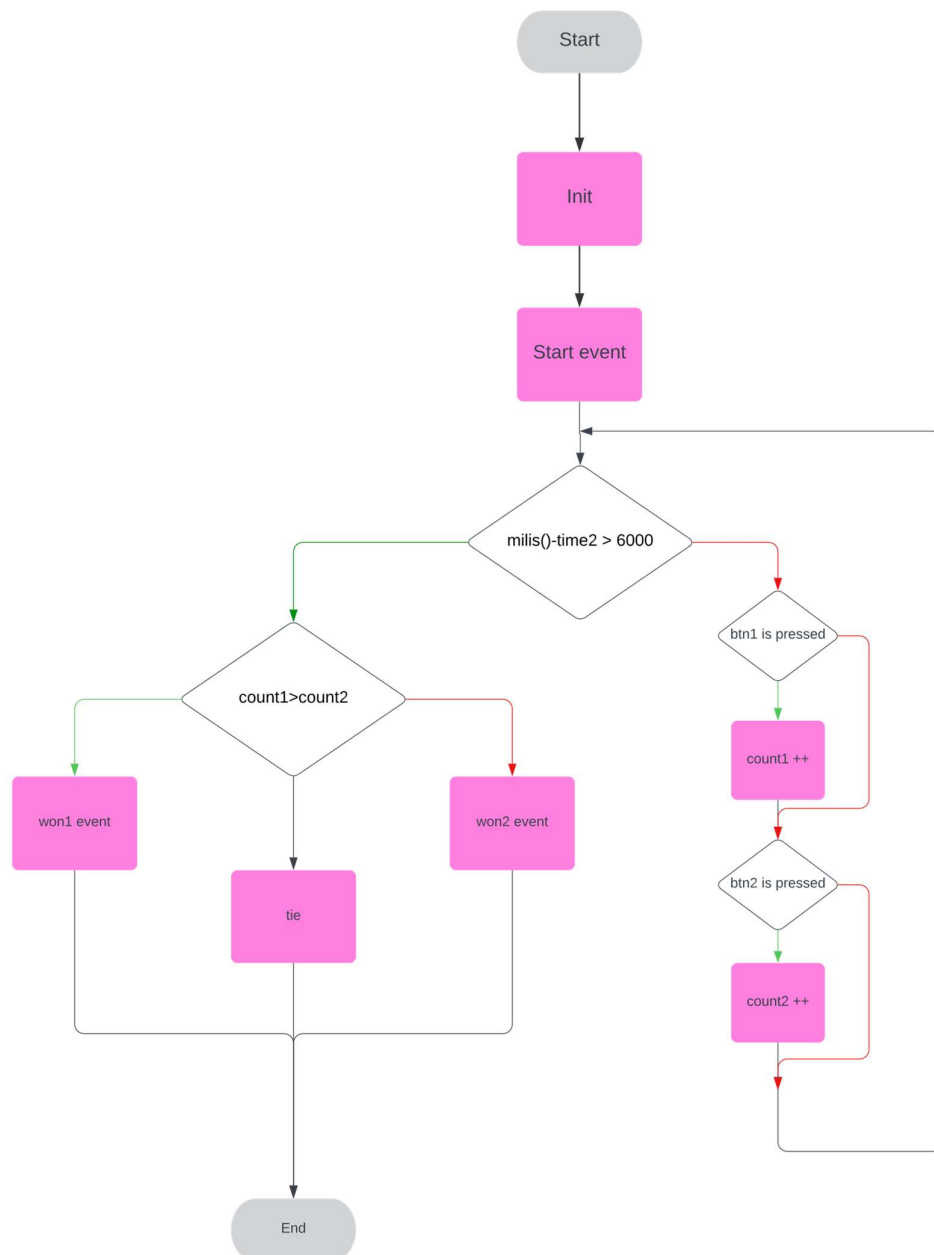
Zadání:

- Vytvořte hru, při které bude za úkol v určitém časovém intervalu zmáčknout tlačítko vícekrát než soupeř.
- Na začátku program automaticky spustí odpočet, vyobrazený na LED diodách
- Po odpočtu se začnou počítat stisknutí tlačítek
- Po skončení časového limitu se vítězi rozsvítí LED diody.

Schéma zapojení:



Vývojový diagram:



Zdrojový kód:

```
#include <stdbool.h>
#include <stm8s.h>
#include <stdio.h>
#include "main.h"
#include "milis.h"
#include "delay.h"
#include "uart1.h"

//definitons for leds on board
//P = pink, B = blue
#define P1_PORT GPIOD
#define P1_PIN GPIO_PIN_3
#define B1_PORT GPIOD
#define B1_PIN GPIO_PIN_1
#define P2_PORT GPIOC
#define P2_PIN GPIO_PIN_3
#define B2_PORT GPIOC
#define B2_PIN GPIO_PIN_2
#define P3_PORT GPIOG
#define P3_PIN GPIO_PIN_0
#define B3_PORT GPIOC
#define B3_PIN GPIO_PIN_1
#define BTN1_PORT GPIOE
#define BTN1_PIN GPIO_PIN_0
#define BTN2_PORT GPIOC
#define BTN2_PIN GPIO_PIN_4
//User macros
#define start_event_duration 3000 //duration of start event in ms
#define DEBOUNCE_TIME 15 //debounce time in ms

void init(void)
{
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); // taktovani MCU na 16MHz
    GPIO_Init(LED_PORT, LED_PIN, GPIO_MODE_OUT_PP_LOW_SLOW);
//Leds init
    GPIO_Init(P1_PORT, P1_PIN, GPIO_MODE_OUT_PP_HIGH_SLOW);
    GPIO_Init(B1_PORT, B1_PIN, GPIO_MODE_OUT_PP_HIGH_SLOW);
    GPIO_Init(P2_PORT, P2_PIN, GPIO_MODE_OUT_PP_HIGH_SLOW);
    GPIO_Init(B2_PORT, B2_PIN, GPIO_MODE_OUT_PP_HIGH_SLOW);
    GPIO_Init(P3_PORT, P3_PIN, GPIO_MODE_OUT_PP_HIGH_SLOW);
    GPIO_Init(B3_PORT, B3_PIN, GPIO_MODE_OUT_PP_HIGH_SLOW);

// Button init
    GPIO_Init(BTN1_PORT, BTN1_PIN, GPIO_MODE_IN_PU_NO_IT);
    GPIO_Init(BTN2_PORT, BTN2_PIN, GPIO_MODE_IN_PU_NO_IT);
    GPIO_Init(BTN_PORT, BTN_PIN, GPIO_MODE_IN_FL_NO_IT);
}
```

```

//GPIO_Init(GPIOE,GPIO_PIN_0 , GPIO_MODE_IN_PU_NO_IT);// btn1
//GPIO_Init(GPIOD,GPIO_PIN_5 , GPIO_MODE_IN_PU_NO_IT);// btn2

//initialize timer and uart
init_milis();
init_uart1();
}

void start_event(void)
{
    //Turn LED on with time delay
    delay_ms(start_event_duration/6);
    GPIO_WriteLow(P1_PORT, P1_PIN);
    delay_ms(start_event_duration/6);
    GPIO_WriteLow(B1_PORT, B1_PIN);
    delay_ms(start_event_duration/6);
    GPIO_WriteLow(P2_PORT, P2_PIN);
    delay_ms(start_event_duration/6);
    GPIO_WriteLow(B2_PORT, B2_PIN);
    delay_ms(start_event_duration/6);
    GPIO_WriteLow(P3_PORT, P3_PIN);
    delay_ms(start_event_duration/6);
    GPIO_WriteLow(B3_PORT, B3_PIN);
}

int main(void){
    uint8_t count1 = 0; // Initialize counter for btn_1
    uint8_t count2 = 0; // Initialize btn2 counter for btn_2
    uint32_t time = 0; // Initialize time
    uint32_t time2 = 0; // Initialize time
    uint32_t last_press_time_btn1 = 0; // Initialize last_press_time_btn1
    uint32_t last_press_time_btn2 = 0; // Initialize last_press_time_btn2

    init();

    bool last_state_btn1 = false;
    bool last_state_btn2 = false;

    start_event();
    HIGH(P1);
    HIGH(P2);
    HIGH(P3);
    HIGH(B1);
    HIGH(B2);
    HIGH(B3);
    time = milis();
    time2 = milis();

```

```

    REVERSE(LED);

    while(1){
        if (milis()-time > 1000){
            time = milis();
            GPIO_WriteReverse(LED_PORT, LED_PIN);
        }
        if (milis()-time2 > 6000){
            break;
        }
        //Detection for btn_1 press

        if(PUSH(BTN1) && !last_state_btn1 && (milis() -
last_press_time_btn1 > DEBOUNCE_TIME)){
            last_state_btn1 = true;
            last_press_time_btn1 = milis();
            count1++;
        }
        else if(!PUSH(BTN1)){
            last_state_btn1 = false;
        }
        //Detection for btn_2 press
        if(PUSH(BTN2) && !last_state_btn2 && (milis() -
last_press_time_btn2 > DEBOUNCE_TIME)){
            last_state_btn2 = true;
            last_press_time_btn2 = milis();
            count2++;
        }
        else if(!PUSH(BTN2)){
            last_state_btn2 = false;
        }
    }

    //Light winners LED

    printf("count1: %d\n \r", count1);
    delay_ms(20);
    printf("count2: %d\n \r", count2);
    delay_ms(20);
    if(count1 < count2){
        for(int i = 0; i < 3; i++){
            LOW(B1);
            LOW(B2);
            LOW(B3);
            delay_ms(250);
            HIGH(B1);
            HIGH(B2);
            HIGH(B3);
            delay_ms(250);
        }
    }

```

```

    }}
else if(count1 > count2){
    for(int i = 0; i < 3; i++){
        LOW(P1);
        LOW(P2);
        LOW(P3);
        delay_ms(250);
        HIGH(P1);
        HIGH(P2);
        HIGH(P3);
        delay_ms(250);
    }
}
else if(count1 == count2){
    for(int i = 0; i < 3; i++){
        LOW(P1);
        LOW(P2);
        LOW(P3);
        LOW(B1);
        LOW(B2);
        LOW(B3);
        delay_ms(250);
        HIGH(P1);
        HIGH(P2);
        HIGH(P3);
        HIGH(B1);
        HIGH(B2);
        HIGH(B3);
        delay_ms(250);
    }
}
}
}

```

```

/*----- Assert -----
*/

```