

Vyšší odborná škola a Střední průmyslová škola elektrotechnická  
Božetěchova 3, Olomouc  
Laboratoře elektrotechnických měření

# Protokol do MIT

Název úlohy

Hudební Nástroj

Číslo úlohy

1

## ZADÁNÍ:

Jednoduchý „hudební nástroj“ realizovaný pomocí maticové klávesnice

Poř. č.

1

Příjmení a jméno

Balog Lukáš

Třída

4B

Skupina

1

Školní rok

2021/22

Datum měření

20. 4. 2022

Datum odevzdání

20. 4. 2022

Počet listů

8

příprava

Klasifikace

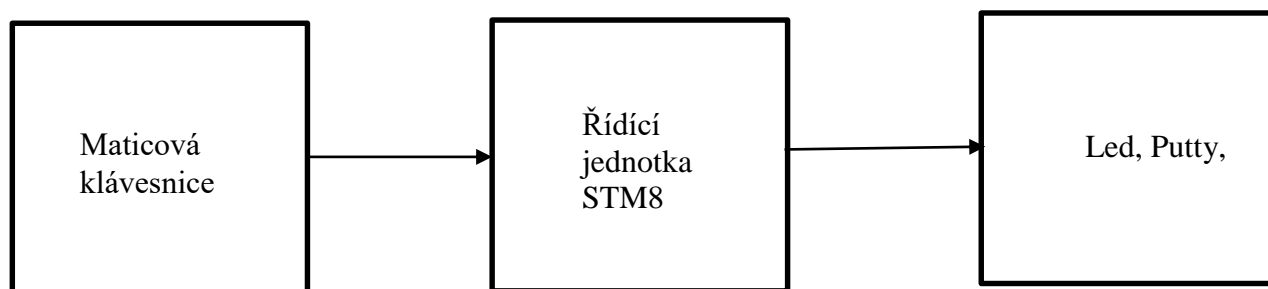
měření

protokol

obhajoba

## Schéma:

Blokové schéma:



## Slovní popis:

### *Program*

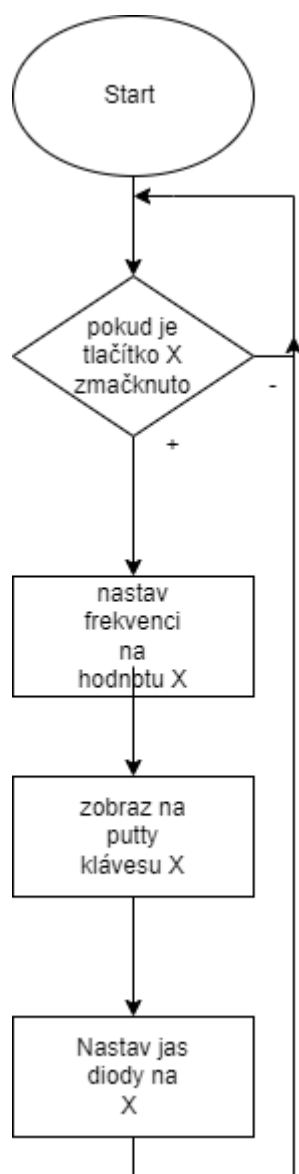
Ovládacím prvkem je maticová klávesnice, po zmáčknutí tlačítka na klávesnici se změní frekvence na buzzeru, tím se změní i vydávaný tón. Další součástí je dioda, jejíž svit je regulován pomocí pwm. Nakonec se ukazuje, které tlačítko bylo zmáčknuto na počítači pomocí Uart a aplikace Putty

### *zapojení*

Piny mikrokontroleru jsou připojeny na maticovou klávesnici a diodu s ochranným odporem

Jméno: Lukáš Balog	Třída: 3B	Číslo protokolu: 201-3R	List: 2/8
--------------------	-----------	-------------------------	-----------

## Vývojový diagram:



## Závěr

Projekt vůbec nedává smysl, ale plní 4 body, abych mohl projít. To je důležité. Doufám, že je to naposledy co musím STM8 použít.

## Kód

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

#define _ISOC99_SOURCE
#define _GNU_SOURCE

void setup(void)
{
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1);           //
    taktovani MCU na 16MHz

    init_milis(); //inicializace mmilisu
    init_uart1();
    init_keypad();
    GPIO_Init(GPIOD, GPIO_PIN_7, GPIO_MODE_OUT_PP_LOW_FAST);

    TIM2_TimeBaseInit(TIM2_PRESCALER_16, 7000 - 1 );
    TIM2_OC1Init(TIM2_OCMODE_PWM1, TIM2_OUTPUTSTATE_ENABLE, 3000,
TIM2_OCPOLARITY_HIGH); // inicializujeme kanál 1 (TM2_CH1)
    TIM2_OC1PreloadConfig(ENABLE);
    TIM2_Cmd(ENABLE);
```

```

}

int main(void)
{
    uint8_t key_last = 0;
    uint8_t key_now = 0;
    uint8_t key_pressed = 0;
    int32_t time_key = 0;

    setup();

    while (1) {

        if (milis() - time_key > 50) {
            time_key = milis();
            key_now = check_keypad();
            if (key_last == 0xFF && key_now != 0xFF) {
                key_pressed = key_now;
                printf("%X\r\n", key_pressed);
            }
            key_last = key_now;
        }

        switch(key_pressed)
        {
            case 1:
                _delay_us(40);
                GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
                TIM2_SetCompare1(7000);
                break;
            case 2:
                _delay_us(50);
                GPIO_WriteReverse(GPIOD, GPIO_PIN_7);

```

```
TIM2_SetCompare1(6500);
break;

case 3:
    _delay_us(60);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(6000);
    break;

case 4:
    _delay_us(70);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(5500);
    break;

case 5:
    _delay_us(80);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(5000);
    break;

case 6:
    _delay_us(90);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(4500);
    break;

case 7:
    _delay_us(100);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(4000);
    break;

case 8:
    _delay_us(110);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(3500);
    break;

case 9:
    _delay_us(120);
```

```
        GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
        TIM2_SetCompare1(3000);
        break;
case 10:
    _delay_us(130);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(2500);
    break;
case 11:
    _delay_us(140);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(2000);
    break;
case 12:
    _delay_us(150);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(1500);
    break;
case 13:
    _delay_us(160);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(1000);
    break;
case 14:
    _delay_us(170);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(500);
    break;
case 15:
    _delay_us(180);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
    TIM2_SetCompare1(200);
    break;
case 0:
    _delay_us(190);
    GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
```

```

        TIM2_SetCompare1(100);
        break;

    default:
        _delay_us(200);
        GPIO_WriteReverse(GPIOD, GPIO_PIN_7);
        TIM2_SetCompare1(50);
        break;
    }
}

/*----- Assert -----
-----*/
#include "__assert__.h"

```

Jméno: Lukáš Balog	Třída: 3B	Číslo protokolu: 201-3R	List: 8/8
--------------------	-----------	-------------------------	-----------