

**Vyšší odborná škola a Střední průmyslová škola elektrotechnická  
Božetěchova 3, Olomouc**

**Laboratoře mikroprocesorové techniky**

# SAMOSTATNÝ PROJEKT MIT

Název úlohy

**MINUTKA**

Číslo úlohy

**MIT-01**

Zadání

1. V levé části displeje se zobrazují minuty (0-99), v pravé části sekundy.
2. Točením enkodéru doprava se přičítá čas po 10s, doleva se čas ubírá. Otočením doleva přes nulu se čas nastaví na maximum.
3. Kliknutím enkodéru se minutky rozběhnou/zastaví
4. Podržením enkodéru na 3s se čas vyresetuje.
5. Po uběhnutí času budou minutky 10s vydávat zvuk a blikat displejem.

Poř. č.

**15**

Příjmení a jméno

**MICHALČÍK Ondřej**

Třída

**4A**

Skupina

**2**

Školní rok

**2021/22**

Datum vypracování

**17. 2. 2022**

Datum odevzdání

**18. 2. 2022**

Počet listů

**5**

Klasifikace

Protokol obsahuje:

Blokové schéma

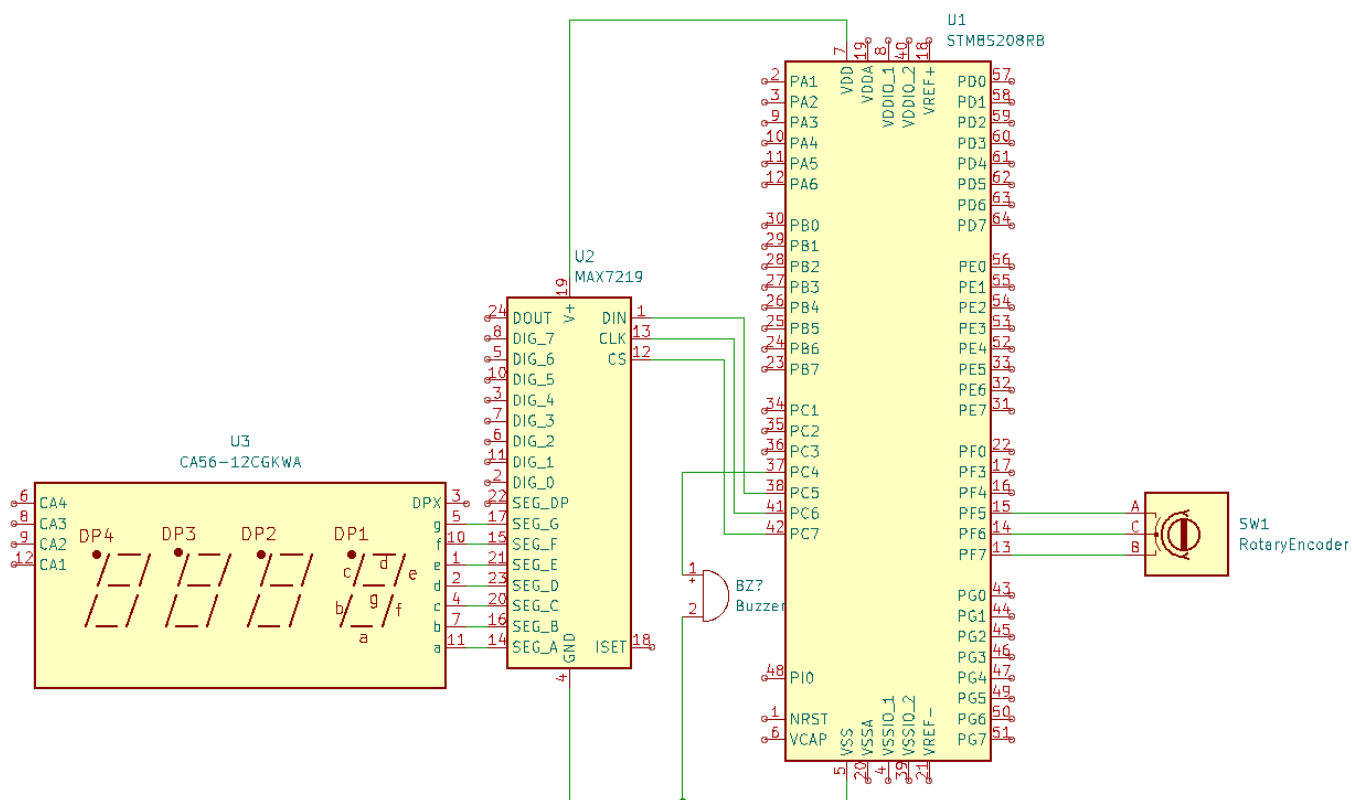
Program

Vývojový diagram

Závěr

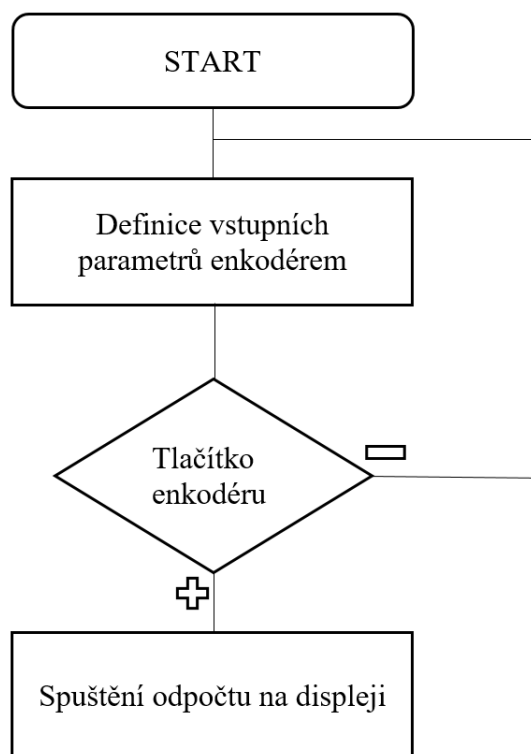
Schéma zapojení

## Zapojení:



Ke kitu STM8 je připojen segmentový displej s driverem MAX7219, rotační enkodér, a bzučák

## Vývojový diagram:



## ZÁVĚR:

Projekt nalezne své využití v každé kuchyni. Pro praktičtější využití by bylo dobré pro něj vyrobit nějaké pouzdro (krabičku). Vypracování celého projektu zabralo 10 až 20 hodin v průběhu několika týdnů.

## Kód:

```
#include "stm8s.h"
#include "milis.h"
#include "swspi.h"
```

```
#define DECODEMODE      (0x9<<8)
#define INTENSITY       (0xa<<8)
#define SCANLIMIT       (0xb<<8)
#define SHUTDOWN        (0xc<<8)
#define DTEST           (0xf<<8)
```

```
#define d1               (0x1<<8)
#define d2               (0x2<<8)
#define d3               (0x3<<8)
#define d4               (0x4<<8)
#define d5               (0x58<<4)
```

Jméno: Ondřej MICHALČÍK	Třída: 4A	Číslo protokolu: MIT-01	List: 3/8
-------------------------	-----------	-------------------------	-----------

```

#define d5b                (0x5<<8)
#define d6                  (0x6<<8)
#define d7                  (0x7<<8)
#define d8                  (0x8<<8)

void max7219_init(void);
void init_enc(void);
void init_timer(void);

//funkce
void klok(void);
void pauza (void);
void had(void);
void displej_sender(void);
void process_enc(void);
void end_anim(void);

volatile int16_t hodnota=0;          //proměnné enkodéru

uint32_t sekunda=0;                 //proměnné pro čas

uint32_t l_snake=0;                  //proměnné pro funkci hada na stranách displeje
uint8_t sneaky=64;

uint16_t l_reset=0;                 //proměnné resetu

uint8_t l_pauza=2;                  //proměnné k pauzování displeje

uint8_t end=0;                      //proměnné pro signalizaci oběhnutí času
uint32_t l_end=0;

void main(void){
CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1);
init_milis();
init_enc();
init_timer();
swspi_init();
max7219_init();
enableInterrupts();
l_snake = milis();

    while(1){
        klok();
        had();
        pauza();
        displej_sender();
    }
}

```

Jméno: Ondřej MICHALČÍK	Třída: 4A	Číslo protokolu: MIT-01	List: 4/8
-------------------------	-----------	-------------------------	-----------

```

void klok (void){
    if (sekunda == 0){l_pauza = 2;}           //aby šel po uběhnutí času opět přičítat enkodérem
}

void had(void){
    if (milis() - l_snake >= 166){
        l_snake = milis();
        sneaky = sneaky>>1;
    }
    if (sneaky == 1){
        sneaky = 64;
    }
}

void pauza (void){
    if (GPIO_ReadInputPin(GPIOF,GPIO_PIN_5) == RESET && l_pauza == 0){
        l_pauza = 1;
        l_reset = milis();
    }
    if (GPIO_ReadInputPin(GPIOF,GPIO_PIN_5) != RESET && l_pauza == 1){
        l_pauza = 2;
    }
    if (GPIO_ReadInputPin(GPIOF,GPIO_PIN_5) == RESET && l_pauza == 2){
        l_pauza = 3;
        l_reset = milis();
    }
    if (GPIO_ReadInputPin(GPIOF,GPIO_PIN_5) != RESET && l_pauza == 3){
        l_pauza = 0;
    }
    if (GPIO_ReadInputPin(GPIOF,GPIO_PIN_5) == RESET && l_pauza == 1 && milis() -
l_reset >= 3000){sekunda = 0;}           //reset při podržení
}

void displej_sender(void){
    swspi_tx16(d1 | sneaky);
    swspi_tx16(d2 | sneaky);
    swspi_tx16(d3 | (sekunda%10));
    swspi_tx16(d4 | (sekunda%60/10));
    swspi_tx16(d5 | (sekunda%600/60));
    swspi_tx16(d6 | (sekunda/600));
    swspi_tx16(d7 | sneaky);
    swspi_tx16(d8 | sneaky);
}

```

```

void end_anim(void){
    l_end = millis();
    if (millis() - l_end <=1000){
        swspi_tx16(DECODEMODE | 0);
        swspi_tx16(d1 | sneaky);
        swspi_tx16(d2 | sneaky);
        swspi_tx16(d3 | sneaky);
        swspi_tx16(d4 | sneaky);
        swspi_tx16(d5b | sneaky);
        swspi_tx16(d6 | sneaky);
        swspi_tx16(d7 | sneaky);
        swspi_tx16(d8 | sneaky);
    }
    if (millis() - l_end >=1001){
        displej_sender();
    }
    if (millis() - l_end >=2000){
        l_end = millis();
    }
}

//odečítání
INTERRUPT_HANDLER(TIM2_UPD_OVF_BRK_IRQHandler, 13){
    TIM2_ClearITPendingBit(TIM2_IT_UPDATE);
    if((l_pauza == 0 || l_pauza == 3) && sekunda>0){sekunda--;}
    pouze při určitých stavech stavového automatu
}

INTERRUPT_HANDLER(TIM3_UPD_OVF_BRK_IRQHandler, 15){
    TIM3_ClearITPendingBit(TIM3_IT_UPDATE);
    process_enc();
}

void max7219_init(void){
    swspi_tx16(DECODEMODE | 0x3c);
    swspi_tx16(INTENSITY | 0x07);
    swspi_tx16(SCANLIMIT | 7);
    swspi_tx16(DTEST | 0);
    swspi_tx16(SHUTDOWN | 1);
}

void init_timer(void){
    TIM3_TimeBaseInit(TIM3_PRESCALER_16,1999);           //timer pro enc
    TIM3_ITConfig(TIM3_IT_UPDATE, ENABLE);
    TIM3_Cmd(ENABLE);
    TIM2_TimeBaseInit(TIM2_PRESCALER_1024,15625);       //timer pro čas
    TIM2_ITConfig(TIM2_IT_UPDATE, ENABLE);
    TIM2_Cmd(ENABLE);
}

```

Jméno: Ondřej MICHALČÍK	Třída: 4A	Číslo protokolu: MIT-01	List: 6/8
-------------------------	-----------	-------------------------	-----------

```

void init_enc(void){
GPIO_Init(GPIOF,GPIO_PIN_7,GPIO_MODE_IN_PU_NO_IT);
GPIO_Init(GPIOF,GPIO_PIN_6,GPIO_MODE_IN_PU_NO_IT);
GPIO_Init(GPIOF,GPIO_PIN_5,GPIO_MODE_IN_PU_NO_IT);
}

void process_enc(void){
uint8_t minule=1;
if(GPIO_ReadInputPin(GPIOF,GPIO_PIN_7) == RESET && minule==1){
minule = 0;
if(GPIO_ReadInputPin(GPIOF,GPIO_PIN_6) == RESET){
if(l_pauza == 1 || l_pauza == 2){sekunda = sekunda-10;}
//zablokuje přidávání při odpočtu
if(sekunda<0){sekunda=5999;}
//odečtení pod nulu nastaví na
maximální čas
}else{
if(l_pauza == 1 || l_pauza == 2){sekunda = sekunda+10;}
if(sekunda>5999){sekunda=0;}
}
}
if(GPIO_ReadInputPin(GPIOF,GPIO_PIN_7) != RESET){minule = 1;}
}

```

```

#ifdef USE_FULL_ASSERT

```

```

/**

```

```

 * @brief Reports the name of the source file and the source line number

```

```

 * where the assert_param error has occurred.

```

```

 * @param file: pointer to the source file name

```

```

 * @param line: assert_param error line source number

```

```

 * @retval : None

```

```

 */

```

```

void assert_failed(u8* file, u32 line)

```

```

{

```

```

/* User can add his own implementation to report the file name and line number,

```

```

ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */

```

```

/* Infinite loop */

```

```

while (1)

```

```

{

```

```

}

```

```

}

```

Jméno: Ondřej MICHALČÍK	Třída: 4A	Číslo protokolu: MIT-01	List: 7/8
-------------------------	-----------	-------------------------	-----------

#endif

Jméno: Ondřej MICHALČÍK	Třída: 4A	Číslo protokolu: MIT-01	List: 8/8
-------------------------	-----------	-------------------------	-----------