Vyšší odborná škola a
Střední průmyslová škola elektrotechnická
Božetěchova 3, 772 00 Olomouc

Samostatný projekt MIT

Název projektu	Číslo projektu
MINUTKY	MIT 1

Zadání

1. Vytvořte zařízení, kde se bude dát nastavit čas. Tento čas se potom začne odpočítávat.

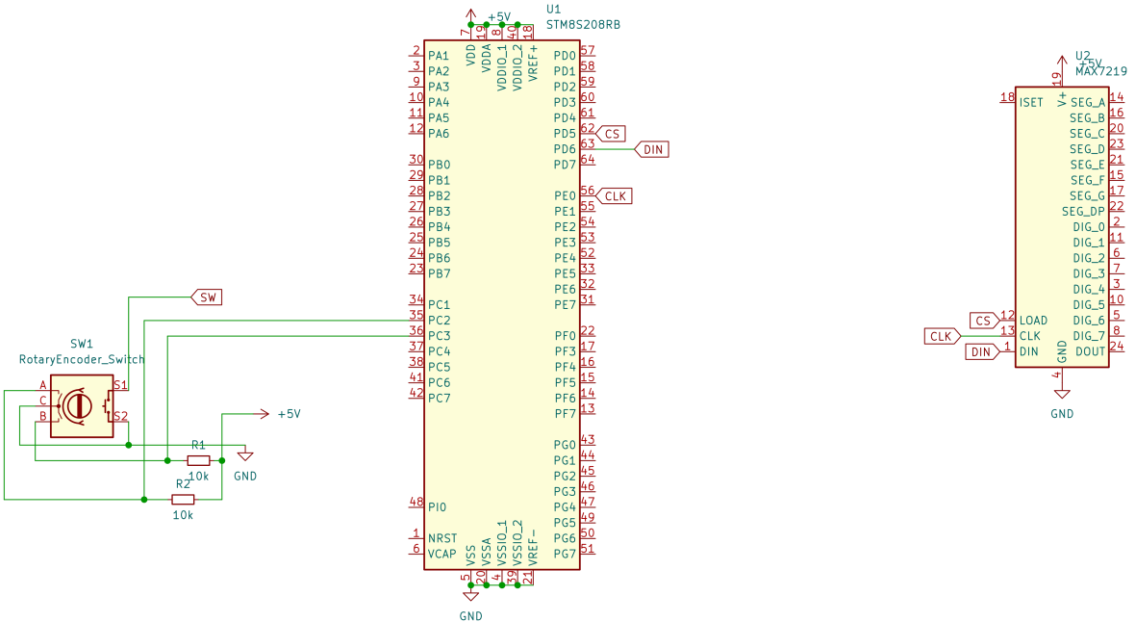
Použitý SOFTWARE: KiCAD, STVD

Celková doba vypracování: 15 hodin

Poř. č.	Příjmení a jméno	Třída	Školní rok
3	BRZOBOHATÝ Bohdan	4A	2021/22

Datum vypracování	Datum odevzdání	Počet listů	Klasifikace
17.1.2022	2.2.2022		

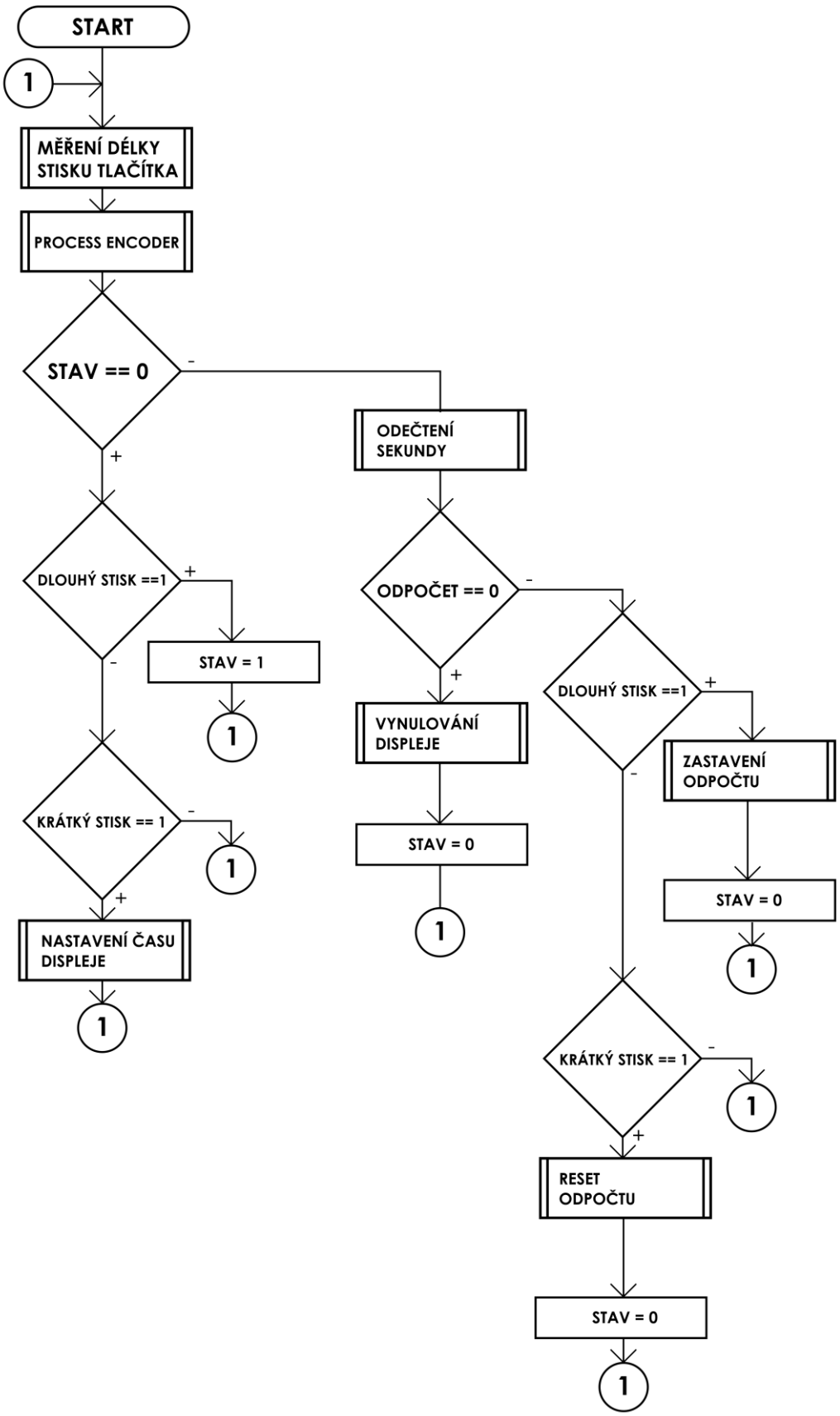
SCHÉMA ZAPOJENÍ:



SLOVNÍ POPIS ZAPOJENÍ:

Mikrokontrolér STM8S208RB řídí chod programu. Pomocí GPIO snímá, zda byl otočen enkodér. Dále je využito tlačítko na samotném kitu pro obsluhu minutek. Jako výstup je připojen pomocí tří pinů displej s driverem MAX7219.

VÝVOJOVÝ DIAGRAM PROGRAMU:



SLOVNÍ POPIS FUNKCE PROGRAMU:

Na začátku programu proběhne inicializace. Při rutině přerušení se zkontroluje stav enkodéru, zda byl otočen. Zároveň při každém průchodu programem se zkontroluje, zda bylo stisknuto tlačítko, případně, jak dlouho bylo stisknuto.

Potom program míří do hlavní podmínky, kde rozlišujeme 2 stavy, V prvním se provádí nastavení odpočítávaného času. V druhém stavu již dochází k odpočtu nastaveného času.

V prvním stavu se potom kontroluje, zda bylo tlačítko stisknuto dlouze, pokud ano, začne odpočet času a přecházíme do stavu 1. Dále se ověří, zda bylo tlačítko stisknuto krátce, pokud ano, mění se funkce enkodéru. V první funkci enkodér volí mezi jednotlivými řády času. V druhé funkci se rotací enkodéru zvyšuje hodnota jednotlivých řádů (hodiny, minuty, sekundy).

Pokud jsme ve druhém stavu, provádí se odpočet. První podmínka zjistí, zda už nedošlo k ukončení odpočtu, pokud ano, vrátí se zpět do prvního nastavovacího času a displej vynuluje. Dále zde má opět různou funkci stisk tlačítka. Při dlouhém stisku přechází program zpět do prvního stavu. Čas na displeji zůstává, lze jej přenastavovat. Pokud v druhém stavu stiskneme tlačítko krátce, dochází k resetování odpočtu, čas se vynuluje a opět je program ve stavu prvním.

ZÁVĚR:

Projekt může být užitečný pro odpočítávání času při běžných situacích, např. Jako odpočet času při úpravě jídla v kuchyni. Tato myšlenka realizace je velice praktická, ale pro lepší zaznamenání konce odpočtu by bylo dobré k zařízení doplnit buzzer pro signalizaci v případě, že člověk displeji nevěnuje plnou pozornost. O tento prvek jsem původně chtěl projekt obohatit, ale nakonec jsem zůstal “pouze” u standardního odpočítávání času, které je i bez zvukové signalizace velice užitečné.

Díky tomuto projektu jsem si rozšířil své znalosti o další možnosti uplatnění STM8 a naučil jsem se pracovat s displejem s driverem MAX7219. Spolu s tímto obohacením jsem své znalosti dále rozšířil ohledně datového typu seznam a naučil jsem se pracovat s enkodérem, který je podle mého názoru tím nejzajímavějším na celém projektu.

MAIN.C:

```
//Bohdan Brzobohatý
#include "stm8s.h"
#include "milis.h"
#include "swspi.h"

//makra pro číslicový displej
#define MAX7219_DIG5 0x6
#define MAX7219_DIG6 0x7
#define MAX7219_DIG7 0x8
#define MAX7219_DECMODE 0x9
#define MAX7219_INTENSITY 0xA
#define MAX7219_SCANLIM 0xB
#define MAX7219_SHUTDOWN 0xC
#define MAX7219_DISTEST 0xF

#define MAX7219_SHUTDOWN_NORMAL_MODE 0b1
#define MAX7219_DECMODE_DIG_ALL 0b11111111
#define MAX7219_DISTEST_OFF 0b0
#define MAX7219_SCANLIM_DIG_ALL 0b111


//piny u enkodéru
#define ENKODER_TLAC_A_GPIO GPIOC
#define ENKODER_TLAC_B_GPIO GPIOC
#define ENKODER_TLAC_GPIO GPIOE

#define ENKODER_TLAC_A_PIN GPIO_PIN_3
#define ENKODER_TLAC_B_PIN GPIO_PIN_2
#define ENKODER_TLAC_PIN GPIO_PIN_4

#define readA
        GPIO_ReadInputPin(ENKODER_TLAC_A_GPIO,ENKODER_TLAC_A_PIN)
#define readB
        GPIO_ReadInputPin(ENKODER_TLAC_B_GPIO,ENKODER_TLAC_B_PIN)


void init_enc(void);
void process_enc(void);
void init_timer(void);
void enc_choice_plus (void);
void enc_choice_minus (void);

uint8_t intensity=1;
uint8_t i;
volatile uint8_t hodnota_enkoderu=0;
volatile bool rotace=0;
volatile bool stisknuto=0,dlouhe_stisknuto=0;
bool konec_stisku=0;
```

```
uint32_t pocatek_stisku_cas=0;
uint32_t odpocet=0;
uint16_t zbytek=0;
volatile bool minule_stisk=0,ted_stisk=0;
volatile bool pocatek_stisku=0;
uint32_t bzucak=0;

uint8_t stav=0;
volatile uint8_t rad=0;
volatile bool vyber=0;
volatile uint8_t cas[3]={
0,0,0
};

void main(void){
CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1);
//inicializace
init_milis();
swspi_init();
init_enc();
init_timer();

//nastavení displeje  - DECODEMODE, SHUTDOWN, DIPLAYTEST, SCANLIMIT,
INTENSISTY
swspi_adressXdata(MAX7219_DECMODE,MAX7219_DECMODE_DIG_ALL);
swspi_adressXdata(MAX7219_SHUTDOWN,MAX7219_SHUTDOWN_NORMAL_MODE
);
swspi_adressXdata(MAX7219_DISTEST,MAX7219_DISTEST_OFF);
swspi_adressXdata(MAX7219_SCANLIM,MAX7219_SCANLIM_DIG_ALL);
swspi_adressXdata(MAX7219_INTENSITY,intensity);

//na poslední dvě pozice nastavíme prázdnou hodnotu
swspi_adressXdata(MAX7219_DIG6,0b1111);
swspi_adressXdata(MAX7219_DIG7,0b1111);

//zapíšeme hodnoty času na displej
swspi_send_time(cas[0],cas[1],cas[2]);


  while (1){
                //při stisknutí tlačítka si zapamatujeme čas
                if(pocatek_stisku==1){
                        pocatek_stisku_cas=milis();
                        pocatek_stisku=0;
                }

                //podle toho, jak dlouho stisk trval, nastavíme jednu z vlajek
                if (konec_stisku==1){
                        if (milis()-pocatek_stisku_cas>999){
                                dlouhe_stisknuto=1;
```

```
                }
                else{
                        stisknuto=1;
                }
                konec_stisku=0;
        }

        //switch pro nastavovací a odpočítající stav
        switch(stav){

                //nastavovací režim
                case 0:
                        //při krátkém stisku program přepíná mezi režimy, kdy se
nastavuje řád a kdy konkrétní hodnota
                        if (stisknuto==1){
                                if (vyber==0){vyber=1;}
                                else{vyber=0;}
                                stisknuto=0;
                        }

                        //při dlouhém stisku se spustí odpočet a přechází se do stavu 1
                        if (dlouhe_stisknuto==1){
                                dlouhe_stisknuto=0;
                                odpocet=milis();
                                zbytek=0;
                                stav=1;
                        }

                        //při výběru řádu (vyber==0) displej jen lehce problikne
                        if (vyber==0){
                                swspi_toggle_slowly(rad,cas[rad]);
                                if(rotace==1){
                                        swspi_send_time(cas[0],cas[1],cas[2]);
                                        rotace=0;
                                }
                        }

                        //při výběru hodnoty na konkrétním řádu (vyber==1) displej
bliká 1:1 (50 % rozsvíceno, 50 % zhasnuto)
                        if (vyber==1){
                                swspi_toggle(rad,cas[rad]);
                                if(rotace==1){
                                        swspi_send_time(cas[0],cas[1],cas[2]);
                                        rotace=0;
                                }
                        }
                        break;

                //odpočítávací režim
                case 1:
```

```
                        //po každé uplynuté sekundě odečteme sekundu
                        if (milis()-(odpocet-zbytek)>999){
                                cas[0]--;//odečtení sekundy
                                if (cas[0]>59){//sekundy přetečou
                                        if (cas[1]!=0 || cas[2]!=0){//a zároveň vyšší řády
nejsou rovny 0
                                                cas[0]=59;//nastavení sekund na 59
                                                cas[1]--;//odečtení minuty
                                                if (cas[1]>59){//přetečení minut
                                                        if (cas[2]!=0){//a zároveň hodiny
nejsou rovny 0
                                                                cas[1]=59;//nastavení minut
na 59
                                                                cas[2]--;//odečtení hodiny
                                                                if (cas[2]>23){//přetečení
hodiny
                                                                        cas[23]=0;//potom
hodiny nastaveny na 0
                                                                }
                                                        }
                                                        else{cas[1]=0;}//jinak minuty=0
                                                }
                                        }
                                        else{cas[0]=0;}//jinak sekundy=0
                                }

                                //ukončení odpočtu
                                if (cas[0]==0 && cas[1]==0 && cas[2]==0){
                                        stav=0;
                                        odpocet=milis();
                                        vyber=0;
                                        rad=0;
                                }
                                odpocet=milis();
                                zbytek=milis()-odpocet;
                                swspi_send_time(cas[0],cas[1],cas[2]);
                        }

                        //při krátkém stisku reset
                        if (stisknuto==1){
                                stisknuto=0;
                                stav=0;
                                vyber=0;
                                rad=0;
                                cas[0]=0;
                                cas[1]=0;
                                cas[2]=0;
                                swspi_send_time(cas[0],cas[1],cas[2]);
                        }
```

```
                                //při dlouhém stisku zastavení odpočtu a přechod do režimu
nastavení
                                if (dlouhe_stisknuto==1){
                                        dlouhe_stisknuto=0;
                                        stav=0;
                                        vyber=0;
                                        rad=0;
                                }
                                break;
                        }
                }
}


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

void init_timer(void){
        TIM3_TimeBaseInit(TIM3_PRESCALER_16,1999);
        TIM3_ITConfig(TIM3_IT_UPDATE, ENABLE);
        TIM3_Cmd(ENABLE);
}


void init_enc(void){
GPIO_Init(ENKODER_TLAC_A_GPIO,ENKODER_TLAC_A_PIN,GPIO_MODE_IN_PU
_NO_IT);
GPIO_Init(ENKODER_TLAC_B_GPIO,ENKODER_TLAC_B_PIN,GPIO_MODE_IN_PU_
NO_IT);
GPIO_Init(ENKODER_TLAC_GPIO,ENKODER_TLAC_PIN,GPIO_MODE_IN_PU_NO_I
T);
}

void process_enc(void){
        static bool minuleA=0;
        static bool minuleB=0;

                if
(GPIO_ReadInputPin(ENKODER_TLAC_GPIO,ENKODER_TLAC_PIN)==RESET){ted_st
isk=1;}
                else{ted_stisk=0;}

                if((ted_stisk==1) && (minule_stisk==0)){pocatek_stisku=1;}
                if(ted_stisk==0 && minule_stisk==1){konec_stisku=1;}

        if((GPIO_ReadInputPin(ENKODER_TLAC_B_GPIO,ENKODER_TLAC_B_PIN) !=
RESET) && minuleB==0 &&
```

```
GPIO_ReadInputPin(ENKODER_TLAC_A_GPIO,ENKODER_TLAC_A_PIN) ==
RESET){
		hodnota_enkoderu--;
		enc_choice_minus();
		rotace=1;
	}
	if((GPIO_ReadInputPin(ENKODER_TLAC_B_GPIO,ENKODER_TLAC_B_PIN)
== RESET) && minuleB==1 &&
GPIO_ReadInputPin(ENKODER_TLAC_A_GPIO,ENKODER_TLAC_A_PIN) != RESET){
		hodnota_enkoderu--;
		enc_choice_minus();
		rotace=1;
	}

	if((GPIO_ReadInputPin(ENKODER_TLAC_A_GPIO,ENKODER_TLAC_A_PIN) !=
RESET) && minuleA==0 &&
GPIO_ReadInputPin(ENKODER_TLAC_B_GPIO,ENKODER_TLAC_B_PIN) ==
RESET){
		hodnota_enkoderu++;
		enc_choice_plus();
		rotace=1;
	}
	if((GPIO_ReadInputPin(ENKODER_TLAC_A_GPIO,ENKODER_TLAC_A_PIN)
== RESET) && minuleA==1 &&
GPIO_ReadInputPin(ENKODER_TLAC_B_GPIO,ENKODER_TLAC_B_PIN) != RESET){
		hodnota_enkoderu++;
		enc_choice_plus();
		rotace=1;
	}

	if(GPIO_ReadInputPin(ENKODER_TLAC_A_GPIO,ENKODER_TLAC_A_PIN) !=
RESET){minuleA = 1;} // pokud je vstup A v log.1
	else{minuleA=0;}
	if(GPIO_ReadInputPin(ENKODER_TLAC_B_GPIO,ENKODER_TLAC_B_PIN) !=
RESET){minuleB = 1;} // pokud je vstup A v log.1
	else{minuleB=0;}

	minule_stisk=ted_stisk;
}

//otočení enkodéru, dojde k odečtení
void enc_choice_minus (void){
	if (stav==0){
		//vyber++;
		if (vyber==0){rad--;}
		if (rad>2){rad=2;}
		if (vyber==1){
			cas[rad]--;
			if (cas[rad]>59){cas[rad]=59;}
			if (cas[2]>23){cas[2]=23;}
```

```
                }
        }
}

//otočení enkodéru, dojde k přičtení
void enc_choice_plus(void){
        if (stav==0){
                if (vyber==0){rad++;}
                if (rad>2){rad=0;}
                if (vyber==1){
                        cas[rad]++;
                        if (cas[rad]>59){cas[rad]=0;}
                        if (cas[2]>23){cas[2]=0;}
                }
        }
}


#ifdef USE_FULL_ASSERT

void assert_failed(u8* file, u32 line)
{
  while (1)
  {
  }
}
#endif
```