|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Vyšší odborná škola a Střední průmyslová škola elektrotechnickáBožetěchova 3, OlomoucLaboratoře elektrotechnických měření | | | | | | | | | | | | | | | |
| **Protokol do MIT** | | | | | | | | | | | | | | | |
| Název úlohy | |  | | | | | | | | | | | Číslo úlohy | |  |
| Minutky | | | | | | | | | | | | | 1 | | |
| **ZADÁNÍ:**  Stopky s led signalizací. nastavování času pomocí enkodéru. Čas se pak zobrazuje na displayi. Po vypršení času se led rozbliká. | | | | | | | | | | | | | | | |
| Poř. č. | Příjmení a jméno | | | |  | | | | Třída | Skupina | | | Školní rok | |  |
| 1 | Balog Lukáš | | | | | | | | 3B | 1 | | | 2020/21 | | |
| Datum měření | | |  | Datum odevzdání | |  | Počet listů |  | Klasifikace | | | | | | |
| 10. 5. 2021 | | | | 15. 5. 2021 | | | 6 | | příprava | | měření | protokol | | obhajoba | |

### **Schéma:**

r

Dioda, Display

STM8

Enkodér

### **Slovní popis:**

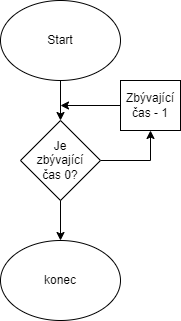
# *Program*

Nastavíme hodnotu v sekundách. Ta se zobrazí na display a po zmáčknutí tlačítka se začne odpočítávat.

Odpočítávání je zobrazeno jako hodnota “zbývající“ na display.

### 

### **Vývojový diagram:**



### **Zhodnocení:**

Pálit čas před maturitou je výsledkem této úlohy z mit. Jakožto dlouhodobkař bych radši dělal něco prospěšného. Třeba přípravu na kpd. Nebo, na ústní z vyt.

Vlastně ani moc nevím, jak kód funguje. Hlavně že ho mám hotový.

**Kód**

#include "stm8s.h"

#include "milis.h"

#include "stm8\_hd44780.h"

#include "delay.h"

void setup(void)

{

    CLK\_HSIPrescalerConfig(CLK\_PRESCALER\_HSIDIV1);      // taktovani MCU na 16MHz

    lcd\_init(); //inicializace LCD

    init\_milis(); //inicializace milis

    GPIO\_Init(GPIOC,GPIO\_PIN\_3,GPIO\_MODE\_OUT\_PP\_LOW\_SLOW); // nastavíme PC5 jako výstup typu push-pull (LEDka)

    GPIO\_Init(GPIOE, GPIO\_PIN\_4,GPIO\_MODE\_IN\_FL\_NO\_IT); // nastavíme PE4 jako vstup (tlačítko)

    GPIO\_Init(GPIOC,

                 ((GPIO\_Pin\_TypeDef)(GPIO\_PIN\_1 | GPIO\_PIN\_2)),

                 GPIO\_MODE\_IN\_PU\_NO\_IT); //inicializace pinů C1 a C2 pro enkoder

    TIM1\_DeInit();

    TIM1\_TimeBaseInit(8, TIM1\_COUNTERMODE\_UP, 60, 8); //inicializace enkoderu

    TIM1\_EncoderInterfaceConfig(TIM1\_ENCODERMODE\_TI12,

                                 TIM1\_ICPOLARITY\_FALLING,

                                 TIM1\_ICPOLARITY\_FALLING);

   TIM1\_Cmd(ENABLE);

}

void signalizace(void) //funkce která bude signalizovat konec odpočtu blikáním ledkou

{

    uint8\_t signalizace\_hodnota = 10; // proměnné pro blikání

    uint32\_t time3 = milis(); // proměnné pro blikání

    while(signalizace\_hodnota){ //každý cyklus se kontroluje jestli uběhlo 300ms pokud ano tak se reverzne pin a odečte se zbývajicí počet kroků

                if (milis() - time3 > 300) {

                    time3 = milis();

                    GPIO\_WriteReverse(GPIOC,GPIO\_PIN\_3);

                    signalizace\_hodnota -= 1;

                }

    }

}

void lcd\_print(unsigned char x\_pos, unsigned char y\_pos, unsigned int value) //funkce pro vypisování čísel na LCD

{

       char tmp[4] = {0x20, 0x20, '\0'} ;

       tmp[0] = (((value / 10) % 10) + 0x30);

       tmp[1] = ((value % 10) + 0x30);

       lcd\_gotoxy(x\_pos, y\_pos);

       lcd\_puts(tmp);

}

int main(void)

{

    uint32\_t time = 0; //inicializace proměnných

    uint32\_t time2 = 0;

    unsigned int present\_value = 0x0000;

    unsigned int previous\_value = 0x0001;

    uint8\_t tlacitko = 0;

    setup();

    lcd\_gotoxy(0,0);

    lcd\_puts("nastaveny");

    lcd\_gotoxy(0,1);

    lcd\_puts("zbyvajici"); //vypsání textu na LCD

    while (1) {

        if(GPIO\_ReadInputPin(GPIOE,GPIO\_PIN\_4)==RESET){ //při stisknutí tlačítka se vezme hodnota co je na enkoderu a pak se spustí stopky s tím časem

            time2 = milis();

            time = present\_value \* 1000;

            tlacitko = 1;

            while(tlacitko){ //loop se opakuje tak dlouho dokud nedojede čas

                lcd\_print(12, 1, ((time2 + time) - milis()) / 1000);

                if (milis() - time2 > time) {

                    signalizace();

                    tlacitko = 0;

                }

            }

        }

//získávání aktuální hodnoty z enkoderu

        present\_value = TIM1\_GetCounter();

        if(present\_value != previous\_value)

        {

            lcd\_print(12, 0, present\_value);

        }

        previous\_value = present\_value;

    }

}

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

#include "\_\_assert\_\_.h"