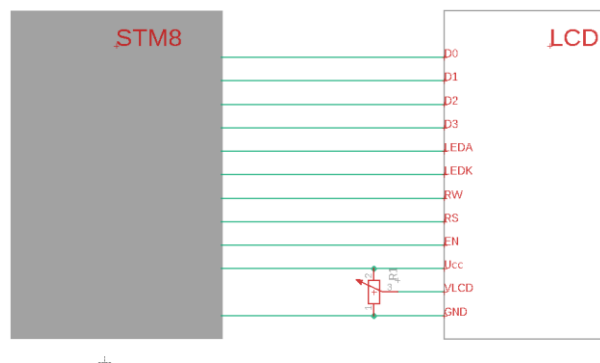


## Popis Projektu

- Vytvořil jsem minutky které počítají a zobrazují na displeji čas nastavený tlačítky, po uplynutí doby zahlásí konec bzučákem
- Softwarová část projektu byla vytvořena v STVD a má 220 řádků

## LCD Displej

- Displej má šestnáct pinů - Datové piny :D0;D1;D2;D3;D4; D5; D6;D7 , Napájecí piny:VSS;VDD;VEE;LEDA;LEDK , Nastavovací piny Rs;Rw,E
- Můj displej má rozměr 16x2, takže jsou 2 sloupce a na každý se vejde 16 znaků

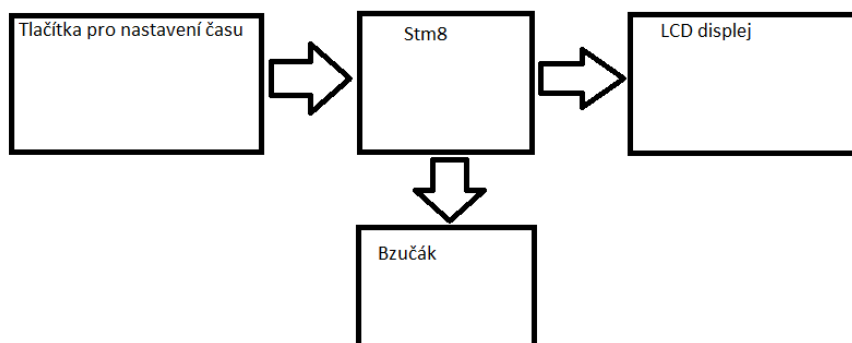


## Bzučák

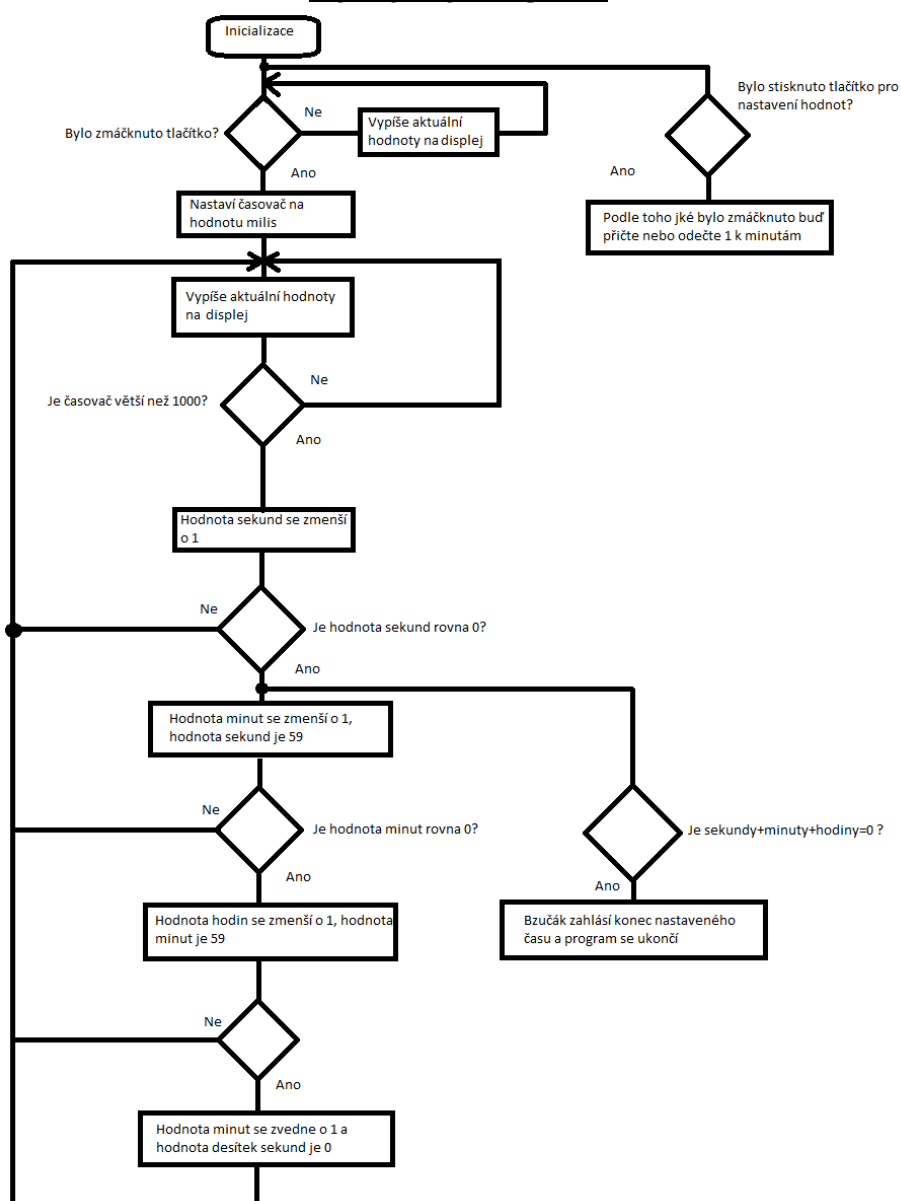
- Zapojení a použití pasivního bzučáku je velice jednoduché
- Můj bzučák má již v sobě vnitřní oscilátor takže do něj nemusím posílat vlastní frekvenci
- Jakmile na výstupu je přivedeno napětí tak začne vydávat zvuk o frekvenci určenou vnitřním oscilátorem



## Blokové schéma



## Vývojový diagram



## Ukázka programu

```
#include "stm8s.h"

#include "milis.h"
#include "stdio.h"
#include "stm8_hd44780.h"
#include "delay.h"

#define NCODER_CLK_PORT GPIOB
#define NCODER_CLK_PIN GPIO_PIN_0
#define NCODER_GET_CLK (GPIO_ReadInputPin(NCODER_CLK_PORT,NCODER_CLK_PIN)!=RESET)

#define NCODER_DATA_PORT GPIOB
#define NCODER_DATA_PIN GPIO_PIN_1
#define NCODER_GET_DATA
(GPIO_ReadInputPin(NCODER_DATA_PORT,NCODER_DATA_PIN)!=RESET)

void setup(void){
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); //Nastavení vstupů a výstupů, rozběhnutí hodin
    init_milis();
    GPIO_Init(LCD_RS_PORT, LCD_RS_PIN,GPIO_MODE_OUT_PP_LOW_SLOW);
    GPIO_Init(LCD_RW_PORT, LCD_RW_PIN,GPIO_MODE_OUT_PP_LOW_SLOW);
    GPIO_Init(LCD_E_PORT,LCD_E_PIN,GPIO_MODE_OUT_PP_LOW_SLOW);

    GPIO_Init(LCD_D4_PORT,LCD_D4_PIN,GPIO_MODE_OUT_PP_LOW_SLOW);
    GPIO_Init(LCD_D5_PORT,LCD_D5_PIN,GPIO_MODE_OUT_PP_LOW_SLOW);
    GPIO_Init(LCD_D6_PORT,LCD_D6_PIN,GPIO_MODE_OUT_PP_LOW_SLOW);
    GPIO_Init(LCD_D7_PORT,LCD_D7_PIN,GPIO_MODE_OUT_PP_LOW_SLOW);

    GPIO_Init(NCODER_CLK_PORT,NCODER_CLK_PIN,GPIO_MODE_IN_FL_NO_IT);
    GPIO_Init(NCODER_DATA_PORT,NCODER_DATA_PIN,GPIO_MODE_IN_FL_NO_IT);

    GPIO_Init(GPIOB,GPIO_PIN_2,GPIO_MODE_OUT_PP_LOW_SLOW);

    GPIO_Init(GPIOE, GPIO_PIN_4,GPIO_MODE_IN_FL_NO_IT);
}

volatile int cisloB = 0;
void main(void){
    char text[32];
    uint16_t n = 0;
    uint32_t timeA = 0;
    uint32_t timeB = 0;
    volatile hodiny = 0;
    volatile minuty = 0;
    volatile sekundy = 0;
    setup();
```

```

lcd_init();
while (1){
    if ((milis() - timeA) > 100){
        timeA = milis();
        if (GPIO_ReadInputPin(NCODER_CLK_PORT,NCODER_CLK_PIN)!=RESET){
            minuty += 1;
            if (minuty > 59){
                hodiny++;
                minuty=0;
                if (hodiny > 23){
                    hodiny=24;
                }
            }
        }
        if (GPIO_ReadInputPin(NCODER_DATA_PORT,NCODER_DATA_PIN)!=RESET){
            if (minuty!=0){
                minuty+= -1;
            }
            if (minuty==0 && hodiny!=0){
                hodiny+= -1;
                minuty=59;
            }
        }
        lcd_gotoxy(0,0);
        sprintf(text,"cas = %2d:%2d:%2d",hodiny,minuty,sekundy);
        lcd_puts(text);
    }
    if ((GPIO_ReadInputPin(GPIOE,GPIO_PIN_4)==RESET)){
        timeB = milis();
        while(1){
            if (milis() - timeB > 1000){
                if ((sekundy) == 0 && (minuty) == 0 && (hodiny) == 0){
                    while (1){
                        GPIO_WriteReverse(GPIOB,GPIO_PIN_2); // zapíšeme na PC5 log. úroveň High (log.1) - rozsvítí LED
                        delay_ms(100);
                    }
                }
                sekundy+=-1;
                if ((sekundy) < 0){
                    if ((minuty) == 0){
                        if ((hodiny)!=0){
                            hodiny+=-1;
                            minuty=60;
                        }
                    }
                    if ((minuty)!=0){
                        minuty+=-1;
                        sekundy=59;
                    }
                }
                timeB=milis();
            }
        }
    }
}

```

```

lcd_gotoxy(0,0);
sprintf(text,"cas = %2d:%2d:%2d",hodiny,minuty,sekundy);
lcd_puts(text);
}
}
}
}

```

```

// pod tímto komentářem nic neměňte
#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)
    {
    }
}
#endif

```

```

/***** (C) COPYRIGHT STMicroelectronics *****/

```

# **Závěr**

## **Sestavení displeje**

- Displej fyzicky sestavit nebylo nic těžkého, byl to spíše problém programování a to mi nedělalo problém
- S displejem sem si neobjednal patici takže sem dráty musel napájet přímo na desku

## **Sestavení Bzučáku**

- Sestavení pasivního bzučáku je velice jednoduché takže mi to nedělalo problémy

## **Sestavení ovládacích tlačítek**

- Tlačítka sem zvolil jako lehčí variantu encoderu protože sem s encoderem narazil na problém na který sem nikde nemohl najít odpověď
- Sestavení a program tlačítek byl velice jednoduchý