

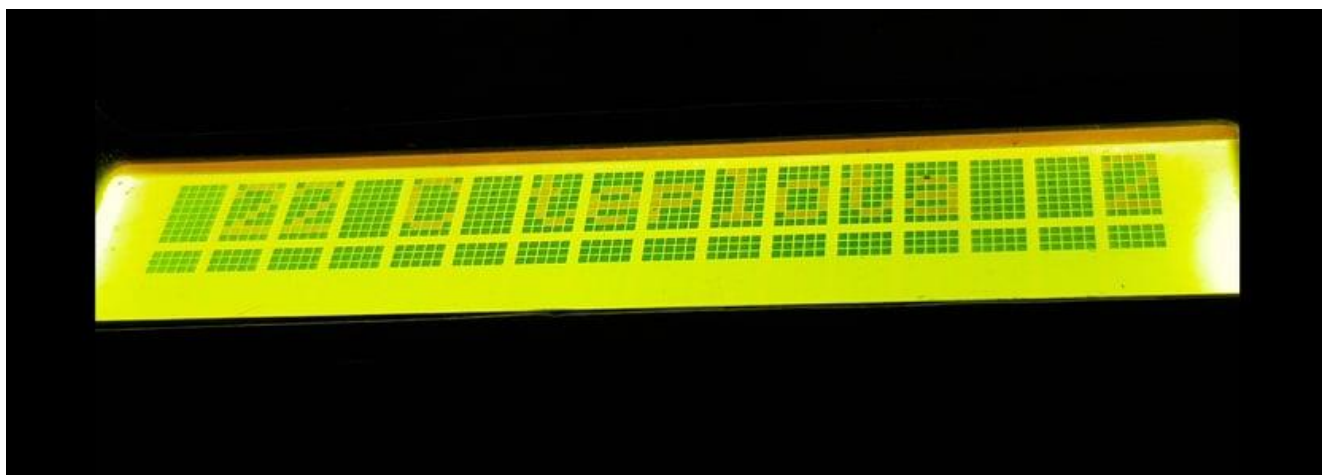
JARNÍ PROJEKT

Výpis teploty na LCD display

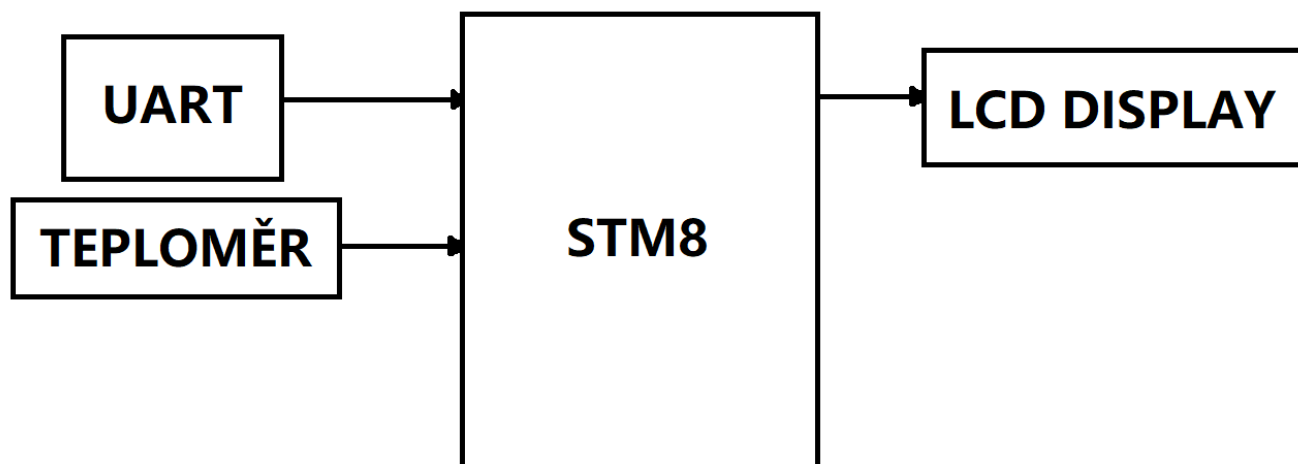
Byl využit ADC, UART, LCD DISPLAY a TEPLOMĚR

Při spuštění se začne na display vypisovat: (teplota) teplota +1/2

1/2 měníme pomocí UARTU který tím určuje zda-li je LED na STM8 ve stavu ON/OFF



Blokové schéma

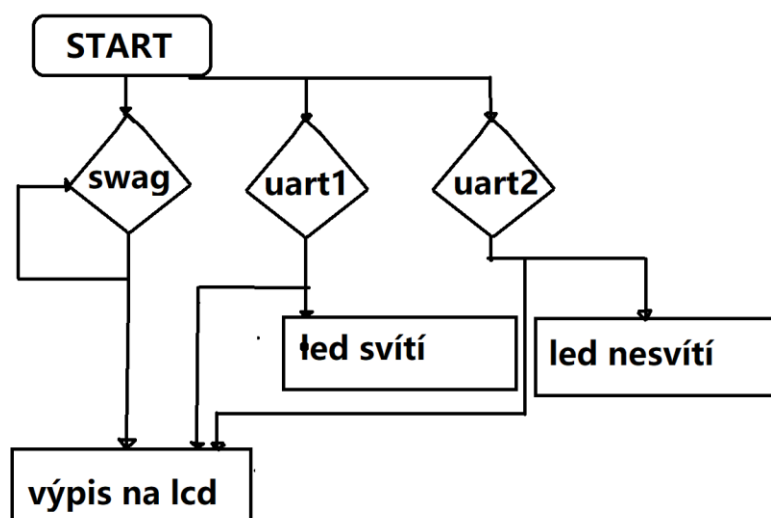


Vývojový diagram

swag ==> milis-last_time >= 500

uart1 ==> příjem z uartu = 1

uart2 ==> příjem z uartu = 2



Program

```
#include "stm8s.h"
#include "milis.h"
#include "spse_stm8.h"
#include "stm8_hd44780.h"
#include "stdio.h"

void ADC_init(void);

uint16_t prevod=0,vysledek=0,y=0;
uint16_t x=0,z=0;
uint16_t last_time=0;
char prijemuart=0;

uint8_t text[32];

void main(void){
CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1);
GPIO_Init(GPIOD,GPIO_PIN_4,GPIO_MODE_OUT_PP_LOW_SLOW);
GPIO_Init(GPIOC,GPIO_PIN_5,GPIO_MODE_OUT_PP_LOW_SLOW);
init_milis();
ADC_init();
lcd_init();
lcd_clear();
UART1_Init(115200,UART1_WORDLENGTH_8D,UART1_STOPBITS_1,UART1_PARITY_NO,UART
1_SYNCMODE_CLOCK_DISABLE,UART1_MODE_RX_ENABLE);
UART1_Cmd(ENABLE);

while (1){

    if(milis()-last_time >= 500){
        last_time=milis();
        y= ADC_get(ADC2_CHANNEL_2);
        vysledek= 5000*y/1024;

        if(UART1_GetFlagStatus(UART1_FLAG_RXNE) != RESET){
//prijmuli jsme byte na uartu
            prijemuart=UART1_ReceiveData8(); //vycteme ho

            sprintf(text,"%c",prijemuart);          //prevedeme char na retezec
                                                    lcd_clear();
                                                    lcd_gotoxy(16,0);//nastavíme
polohu
                                                    lcd_puts(text);//na LCD
nahrajeme řetězec
            if(prijemuart == '1'){
                GPIO_WriteLow(GPIOC,GPIO_PIN_5);
                                                    }
                if(prijemuart == '2'){
                GPIO_WriteHigh(GPIOC,GPIO_PIN_5);
                                                    }
                                                    }

            sprintf(text,"%s","C teplota");//prevedeme text na řetězec
            lcd_gotoxy(4,0);
            lcd_puts(text);

            sprintf(text,"%u",vysledek);//prevedeme číslo na řetězec
            lcd_gotoxy(1,0);
            lcd_puts(text);
```

```

    }

}

void ADC_init(void){
// na pinech/vstupech ADC_IN2 (PB2) a ADC_IN3 (PB3) vypneme vstupní buffer
ADC2_SchmittTriggerConfig(ADC2_SCHMITTTTRIG_CHANNEL2,DISABLE);
ADC2_SchmittTriggerConfig(ADC2_SCHMITTTTRIG_CHANNEL3,DISABLE);
// nastavíme clock pro ADC (16MHz / 4 = 4MHz)
ADC2_PrescalerConfig(ADC2_PRESSEL_FCPU_D4);
// volíme zarovnání výsledku (typicky vpravo, jen vyjmečně je výhodné vlevo)
ADC2_AlignConfig(ADC2_ALIGN_RIGHT);
// nasatvíme multiplexer na některý ze vstupních kanálů
ADC2_Select_Channel(ADC2_CHANNEL_2);
// rozběhneme AD převodník
ADC2_Cmd(ENABLE);
// počkáme než se AD převodník rozběhne (~7us)
ADC2_Startup_Wait();
}

```

Závěr

Program nebylo moc těžké napsat, nejdéle mi zabral asi hodinu času 1 error, kde mi to vypisovalo nesmysly. Nepřepsal jsem %s na %c, když jsem využíval výpis charakteru na lcd display. Celkem hloupá chyba.

Jinak mi to celkově zabralo pár hodin. S výsledkem jsem spokojen. Naučil jsem se využívat nové funkce a periferie.