

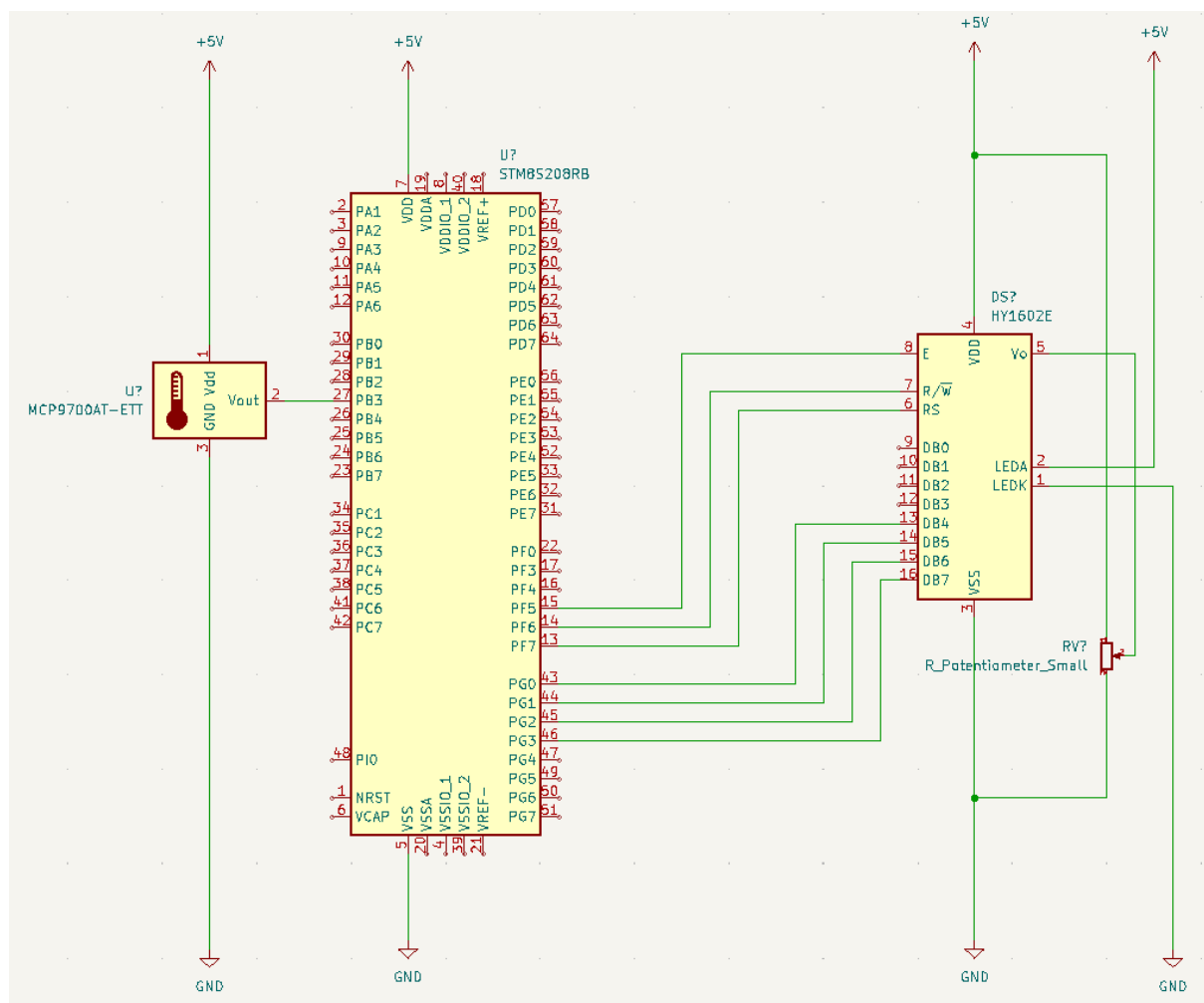
Měřič teploty

Funkce programu:

Můj program slouží jako měřič teploty. Při zapnutí se automaticky začne měřit teplota pomocí teploměru.

Aktuální naměřená hodnota se poté zobrazí přímo na LCD displeji. Díky A/D převodníku se analogové hodnoty převedou na digitální. S těmito hodnotami (číslí) poté už pracuje program například v rámci milis. V zapojení používám také potenciometr, avšak pouze pro úpravu jasu displeje.

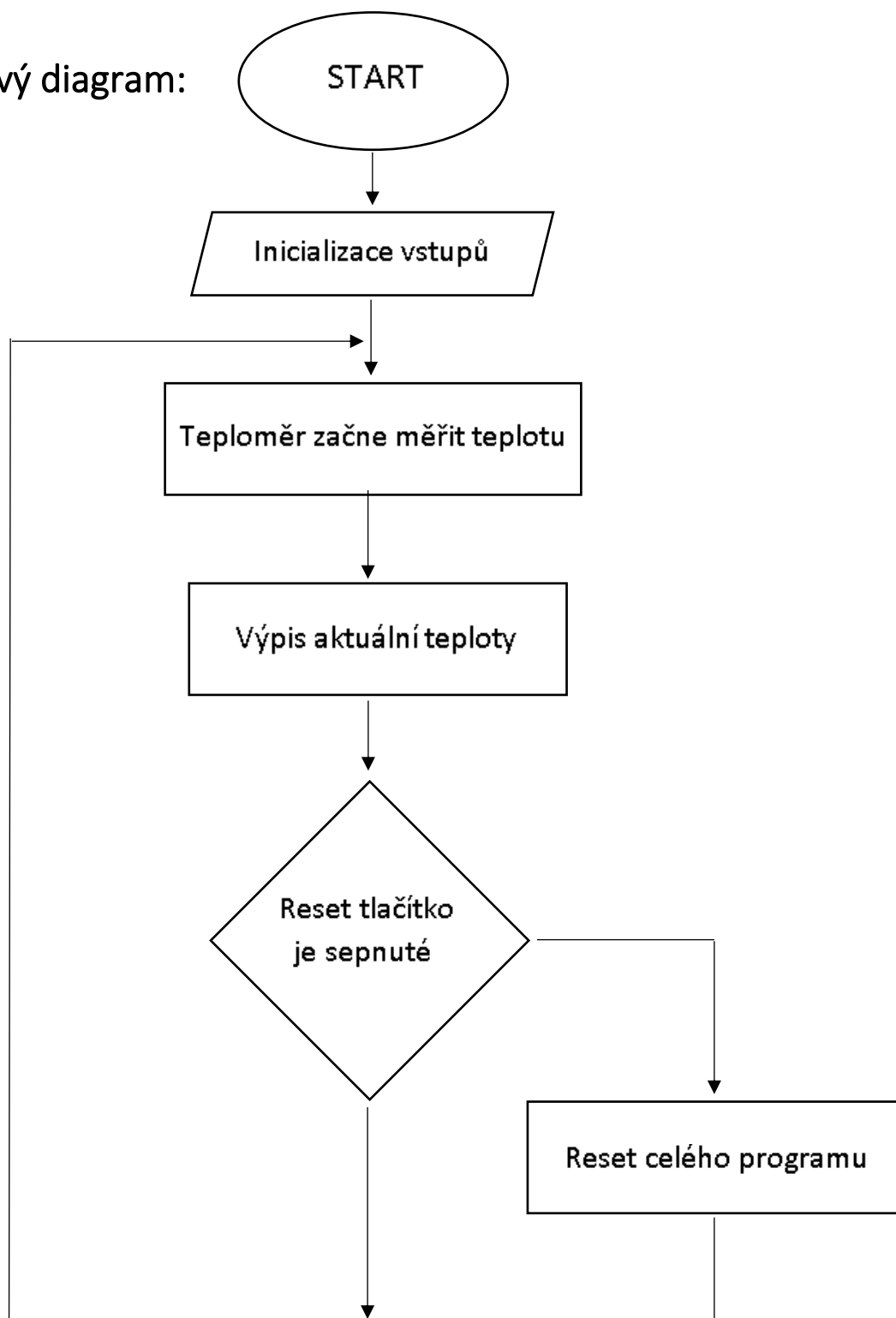
KiCad schéma:



V KiCad schématu můžeme vidět zapojení modulu STM8S, LCD displeje, teploměru a potenciometru. Společné zapojení součástek tvoří funkční obvod.

STM8S je srdcem obvodu, teploměr pro měření okolní teploty, displej pro zobrazení naměřených hodnot a potenciometr pro nastavování jasu.

Vývojový diagram:



Zhodnocení:

Výhodou mého projektu je určitě přínos nových zkušeností v rozhraní mikroprocesoru STM8S a jeho programování. Dále nyní dokážu pracovat s teploměrem/ad převodníkem. Myslím se, že mnou vyrobené zařízení pro měření okolní teploty je skvělým završením čtvrtého ročníku a mikroprocesorové techniky. Nejedná se samozřejmě o složitý projekt. Cílem bylo dovršení sedmi povinných bodů.

Jak to vidím dál?

Rád bych studoval na vysoké škole, ovšem mířím mimo elektrotechniku. Do mikroprocesorové techniky jsem chodil celkem rád, škoda, že znalosti zde získané už nepoužiji.

Zdrojový kód – main.c

```
#include "stm8s.h"
```

```
#include "milis.h"
```

```
#include "stm8_hd44780.h"
```

```
#include <stdio.h>
```

```
#include "delay.h"
```

```
#include "spse_stm8.h"
```

```
#define _ISOC99_SOURCE
```

```
#define _GNU_SOURCE
```

```
void ADC_init(void){
```

```
ADC2_SchmittTriggerConfig(ADC2_SCHMITTTTRIG_CHANNEL3,DISABLE);
```

```
ADC2_PrescalerConfig(ADC2_PRESSEL_FCPU_D4);
```

```
ADC2_AlignConfig(ADC2_ALIGN_RIGHT);  
ADC2_Select_Channel(ADC2_CHANNEL_3);  
ADC2_Cmd(ENABLE);  
ADC2_Startup_Wait();  
}
```

```
void setup(void)  
{  
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1);    // taktovani MCU na  
    16MHz  
    init_milis();  
  
    lcd_init();  
    ADC_init();  
  
}
```

```
int main(void)  
{  
    uint32_t time = 0;  
    uint16_t adc_hodnota;  
    uint16_t teplota;  
    char text[32];  
    setup();
```

```

while (1) {
    if (milis()-time>1000){
        time = milis();

        adc_hodnota = ADC_get(ADC2_CHANNEL_3); // do adc_value ulož
        výsledek převodu vstupu ADC_IN2 (PB2)

        teplota = adc_hodnota * 500;
        teplota = teplota / 1024;

        lcd_gotoxy(0, 0);
        sprintf(text, "Teplota = %u C", teplota);
        lcd_puts(text);
    }
}

/*----- Assert -----*/
#include "__assert__.h"

```