

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

Projektová úloha MIT

Název úlohy

Teploměr

Číslo úlohy

ZADÁNÍ:

1. Vytvořte teploměr, který bude čidlem brát napětí vůči zemi v mV a zobrazovat ho pomocí AD převodníku, který mV změní na $^{\circ}\text{C}$ - $300\text{mV} = 30^{\circ}\text{C}$ a poté to další část programu zobrazí na displayi.

Poř. č.

30

Příjmení a jméno

Závodský Ondřej

Třída

4B

Skupina

2

Školní rok

2020/21

Datum zadání

. . 2022

Datum odevzdání

. . 2022

Počet listů

6

Klasifikace

Textová část

Obhajoba

Funkčnost -

Protokol obsahuje: teoretický úvod

schéma

závěr

TEORETICKÝ ÚVOD:

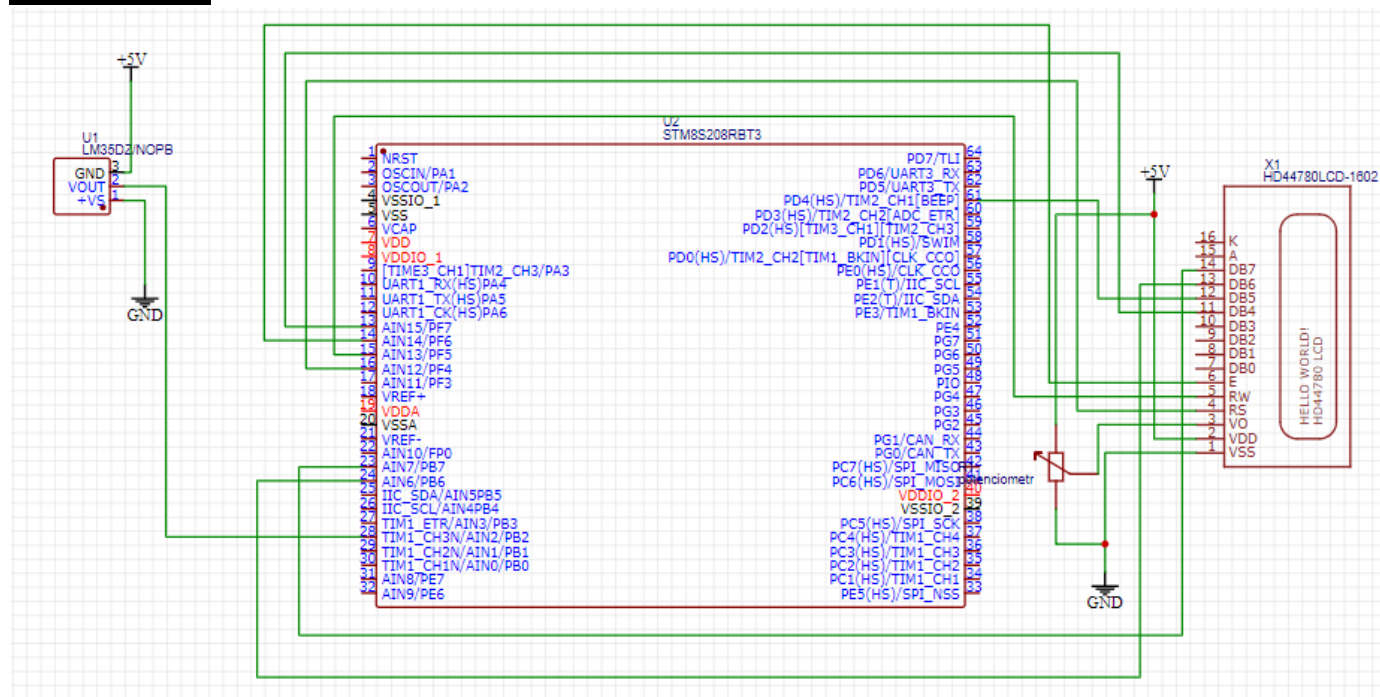
Slovní popis zapojení

Jako projekt jsem si vybral teploměr, který je realizován pomocí čidla LM35DZ na měření teploty. Zem teplotního čidla je zapojena na mínus, jeho napájení je zapojeno na +5V a jeho výstup je zapojen na pin PB2. Potenciometr je zapojený na zem, na +5V a prostřední nožička je zapojena na V0 na displeji a potenciometr je využíván jako trimr. Na displeji jsou vstupy zapojeny RS na PF4, RW na PF5, E na PF6, D4 na PF7, D5 na PD4, D6 na PB6, D7 na PB7. Teplotní čidlo LM35DZ mění napětí v mV na °C - př. 300mV = 30°C a poté se °C zobrazují na displej, který je zapojen na stm8.

Slovní popis funkce

Teplotní čidlo LM35DZ vezme napětí na zemi a pomocí AD převodníku se toto napětí změní na °C a poté se zobrazí na displeji. Kvůli přesnosti čidla se teplota může vychýlit až 0,5 °C, což by ale neměl být žádný problém.

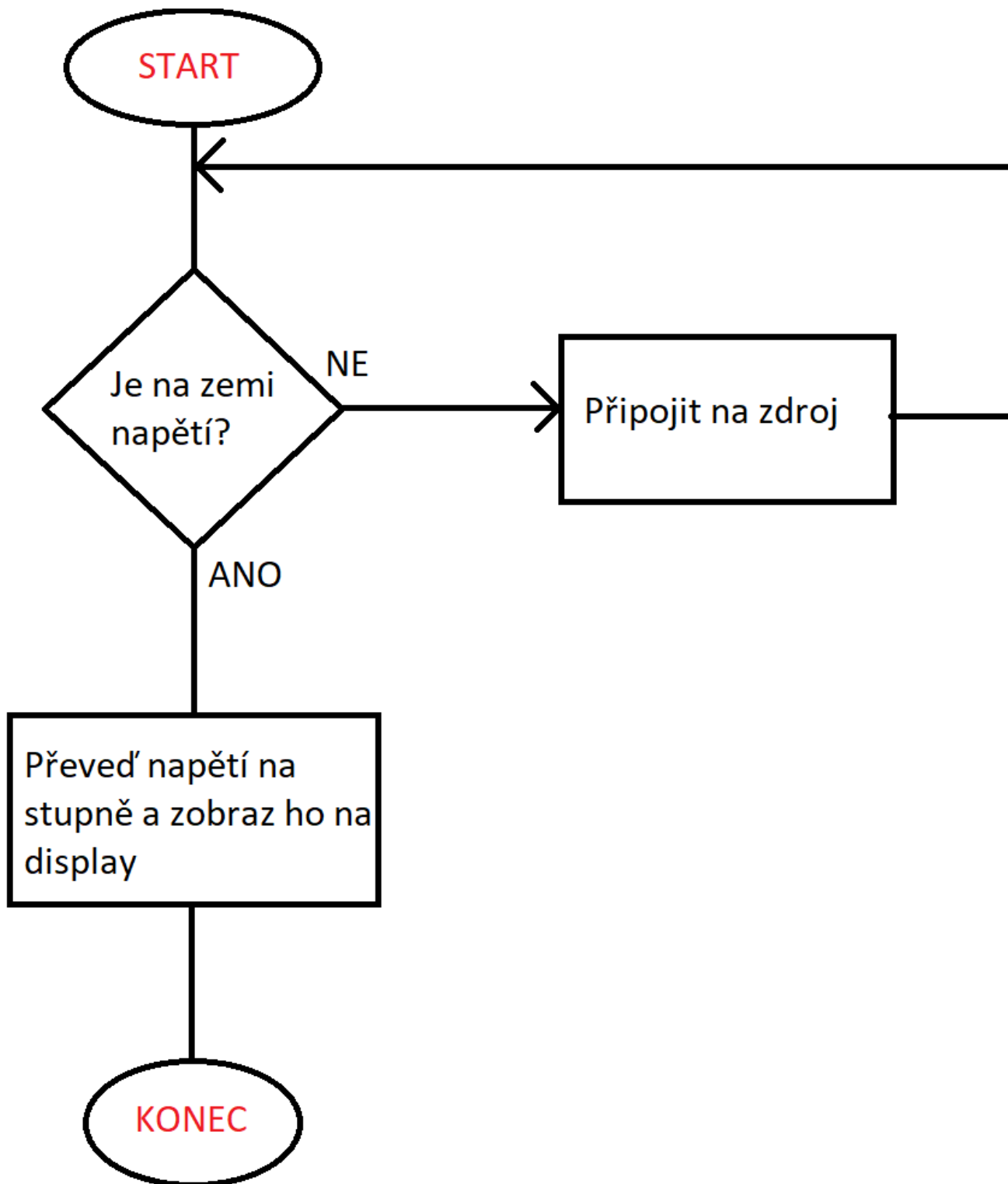
SCHÉMA:



Obrázek 1 - Schéma č. 1: Schéma zapojení tepelného čidla

VÝVOJOVÝ DIAGRAM:

Jméno: Ondřej Závodský	Třída: 3B	Číslo úlohy: 4	List: 2/6
------------------------	-----------	----------------	-----------

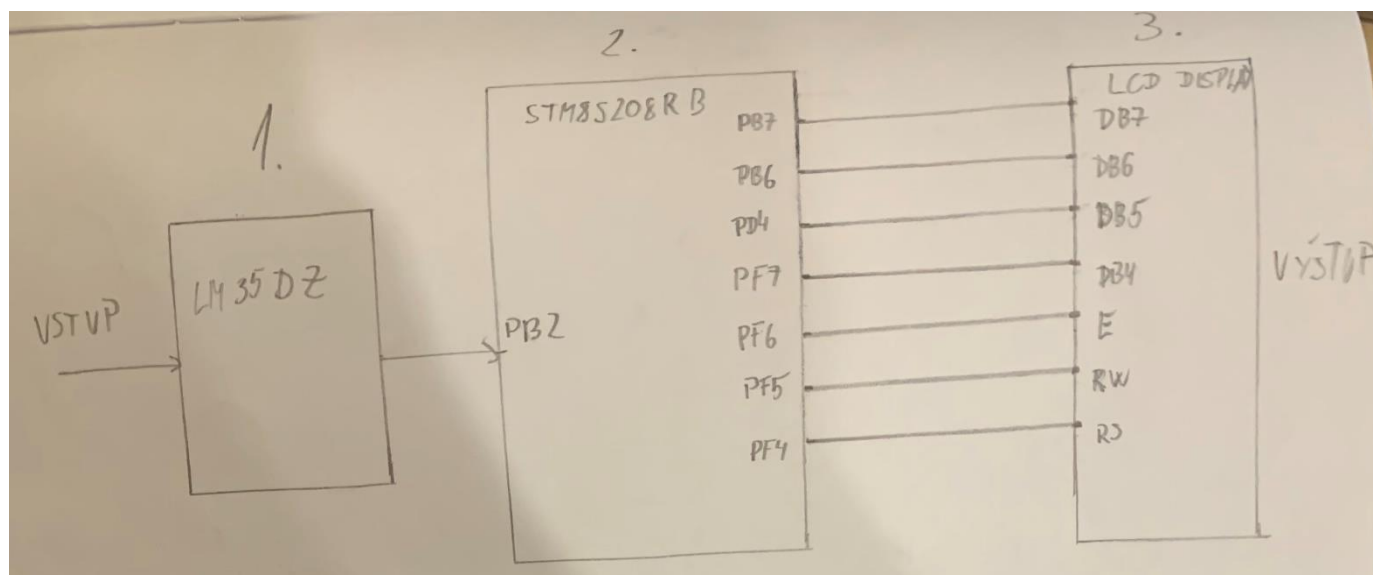


Obrázek 2 - Vývojový diagram č. 1: Vývojový diagram programu

Blokové schéma

Jméno: Ondřej Závodský	Třída: 3B	Číslo úlohy: 4	List: 3/6
------------------------	-----------	----------------	-----------

LM35DZ - změří teplotu na zemi a v mV ji pošle do STM8 s chybou asi 50mV
 STM8S - převede teplotu z mV na stupně $10\text{mV} = 1^\circ\text{C}$ a pošle ji na display
 LCD Display - zobrazí digitálně teplotu ve stupních



Obrázek 3 - Blokové schéma obvodu

ZÁVĚR:

Na projektu se mi pracovalo mnohem líp než minulý rok. Řekl bych, že hodně pomohlo, že jsme používali VSCodium místo STVD a také, že už jsem z minulého roku věděl, jak co funguje. Hodně se mi líbilo jak VSCodium označuje chyby, díky tomu jsem vždycky zjistil, co bylo špatně a na jakém řádku to bylo. Tento projekt určitě hodnotím líp, než ten z minulého roku, ale i tak mě programování nebaví, takže bych radši dělal jiné věci.

main.c:

```
#include "stm8s.h"
#include "milis.h"
#include "stm8_hd44780.h"
#include "spse_stm8.h"
#include "delay.h"
#include <stdio.h>
#define _ISOC99_SOURCE
#define _GNU_SOURCE

void ADC_init(void){
```

```

// na pinech/vstupech ADC_IN2 (PB2) a ADC_IN3 (PB3) vypneme vstupní buffer
ADC2_SchmittTriggerConfig(ADC2_SCHMITTTTRIG_CHANNEL2,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();
}

void TIM2_setup(void){
    TIM2_DeInit();
    TIM2_TimeBaseInit(TIM2_PRESCALER_1, 640 - 1); //25kHz
    TIM2_OC1Init(TIM2_OCMODE_PWM1, TIM2_OUTPUTSTATE_ENABLE, 200,
        TIM2_OCPOLARITY_LOW);
    TIM2_Cmd(ENABLE);
}

void init(void)
{
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); // taktovani MCU na 16MHz
    init_milis();
    lcd_init();
    ADC_init();
    TIM2_setup();
}

int main(void)
{
    init();
    uint32_t timeA = 0;
    uint16_t adc_value;
    char text[32];
    int16_t teplota;
    int16_t teplota_1_cast;
    int16_t teplota_2_cast;

    while (1) {
        if (milis()-timeA>100){
            timeA=milis();
            adc_value = ADC_get(ADC2_CHANNEL_2); // do adc_value ulož výsledek převodu
vstupu ADC_IN2 (PB2)
            teplota = ((uint32_t)adc_value*5000 + 512)/1024;
            teplota_1_cast=teplota/10; //celočíselná část

```

```
        teplota_2_cast=teplota%10;//desetinná část
    }
    lcd_gotoxy(0, 0);
    sprintf(text,"Teplota = %2u %1u C",teplota_1_cast,teplota_2_cast);
    lcd_puts(text);
}
}

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