|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vyšší odborná škola a Střední průmyslová škola elektrotechnická  Božetěchova 3, Olomouc | | | | |
| **Projektová úloha MIT** | | | | |
| Název úlohy | | | | Číslo úlohy |
| Teploměr | | | |  |
| **ZADÁNÍ:**    1. Vytvořte teploměr, který bude čidlem brát napětí vůči zemi v mV a zobrazovat ho pomocí AD převodníku, který mV změní na °C - 300mV = 30°C a poté to další část programu zobrazí na displeyi. | | | | |
| Poř. č. Příjmení a jméno  30 Závodský Ondřej | | | Třída Skupina Školní rok 4B 2 2020/21 | |
| Datum zadání | Datum odevzdání | Počet listů | Klasifikace | |
| . . 2022 | . . 2022 | 6 | Textová část Obhajoba Funkčnost - | |
| Protokol obsahuje: teoretický úvod schéma  závěr | | | | |

**TEORETICKÝ ÚVOD:**

**Slovní popis zapojení**

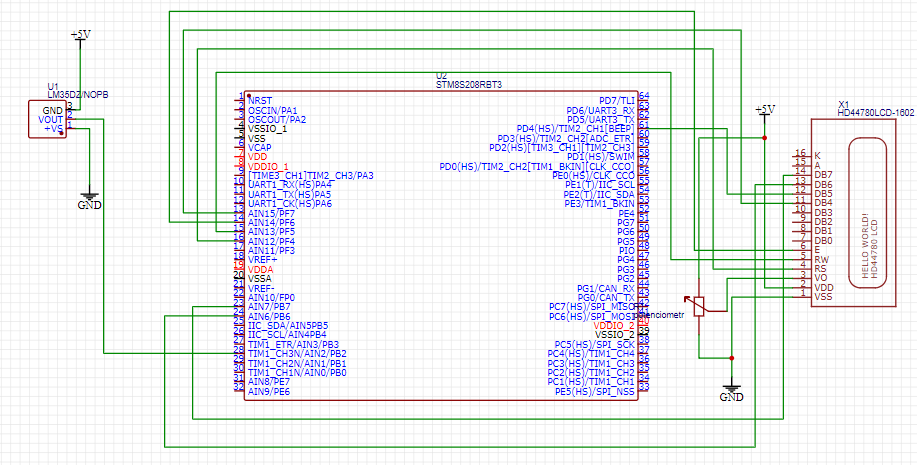
Jako projekt jsem si vybral teploměr, který je realizován pomocí čidla LM35DZ na měření teploty. Zem teplotního čidla je zapojena na mínus, jeho napajéní je zapojeno na +5V a jeho výstup je zapojen na pin PB2. Potenciometr je zapojený na zem, na +5V a prostřední nožička je zapojena na V0 na displeyi a potenciometr je využíván jako trimr. Na displeyi 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 displey, 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 displeyi. Kvůli přesnosti čidla se teplota může vychýlit až

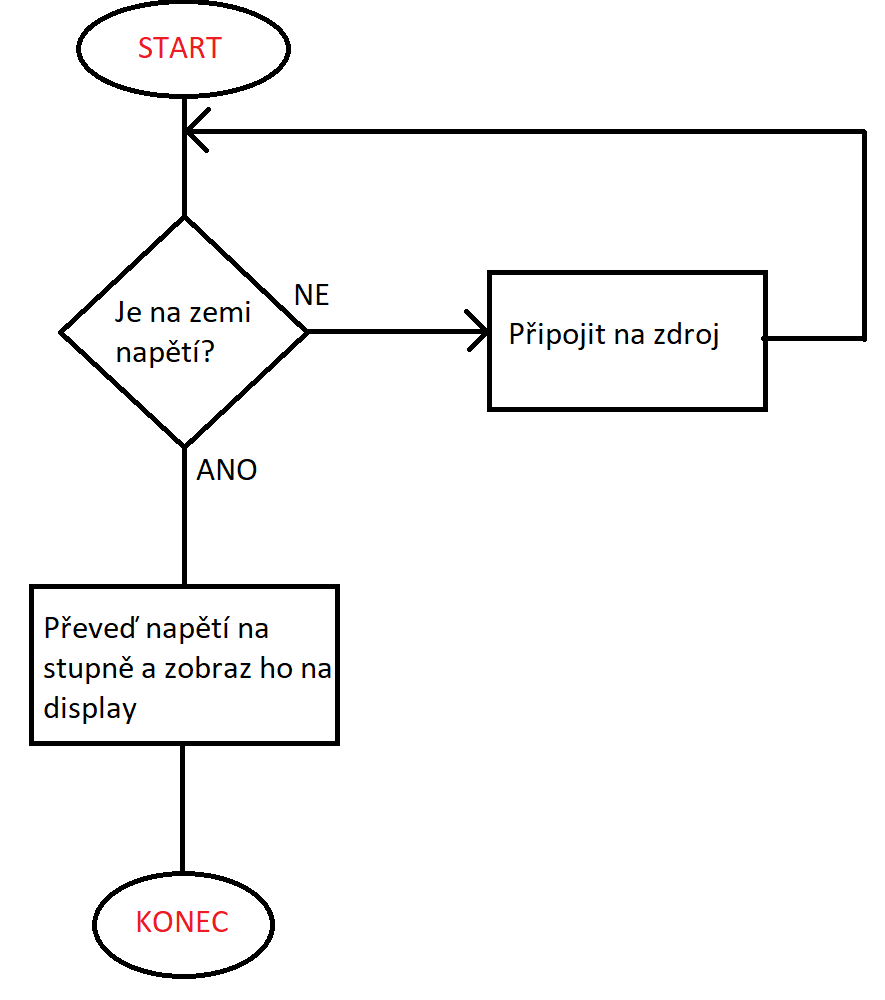
0,5 °C, což by ale neměl být žádný problém.

**SCHÉMA:**



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

**VÝVOJOVÝ DIAGRAM:**



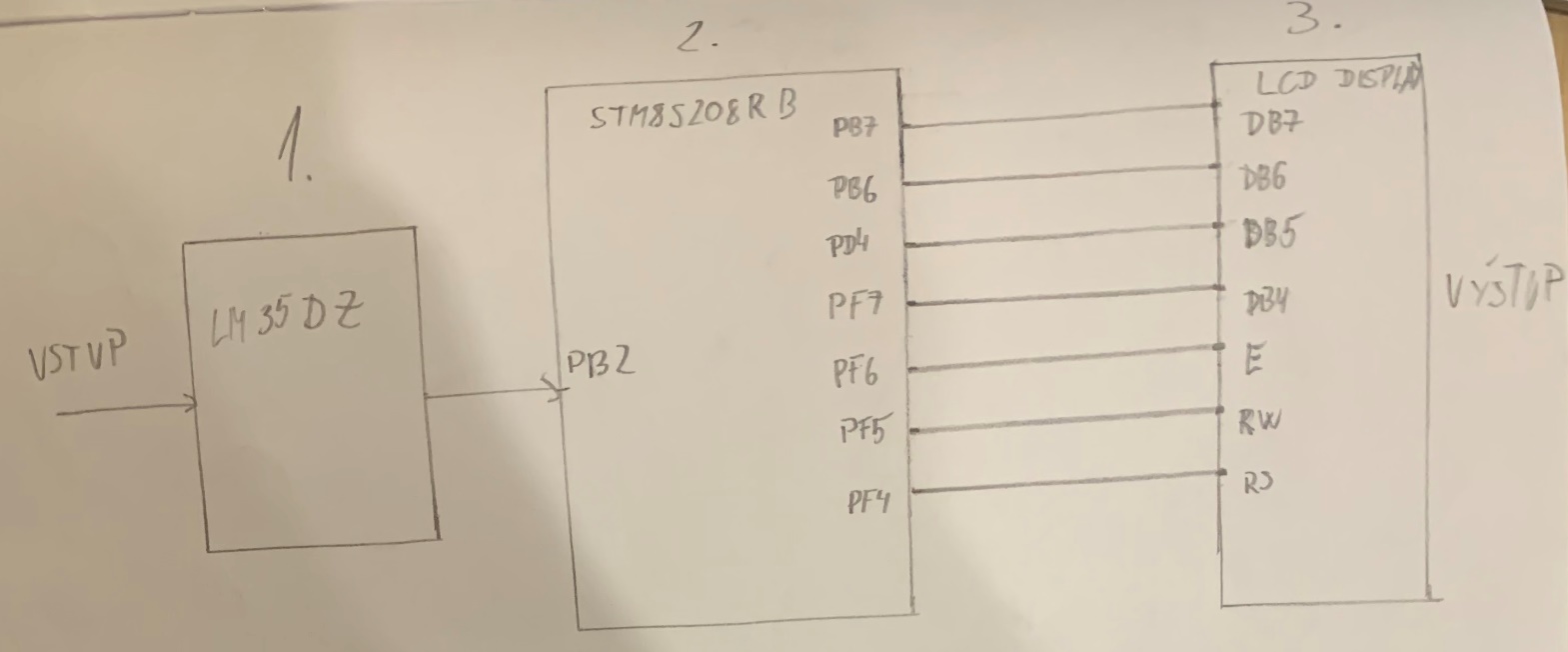
# Obrázek 2 - Vývojový diagram č. 1: Vývojový diagram programu

**Blokové schéma**

LM35DZ - změří teplotu na zemi a v mV ji pošle do STM8 s chybou asi 50mV

STM8S - převede teplotu z mV na stupně 10mV = 1 °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\_SCHMITTTRIG\_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 init(void)

{

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

    init\_milis();

    lcd\_init();

    ADC\_init();

}

int main(void)

{

    init();

    uint32\_t timeA = 0;

    uint16\_t adc\_value;

    char text[32];

    int16\_t voltage1;

    int16\_t teplota1;

    int16\_t teplota2;

    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)

            voltage1 = ((uint32\_t)adc\_value\*5000 + 512)/1024;

            teplota1=voltage1/10;

            teplota2=voltage1%10;

        }

        lcd\_gotoxy(0, 0);

        sprintf(text,"Teplota = %2u %1u C",teplota1,teplota2);

        lcd\_puts(text);

    }

}

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

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