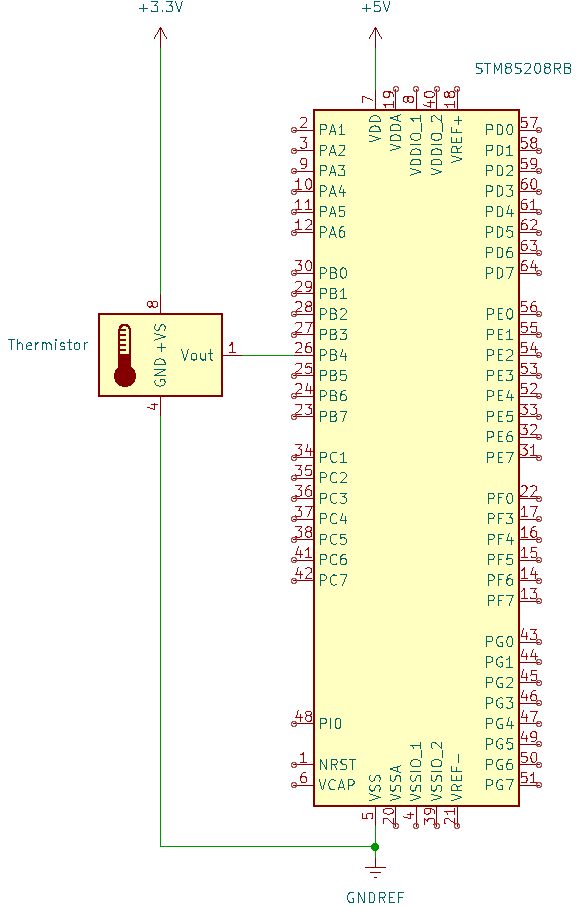
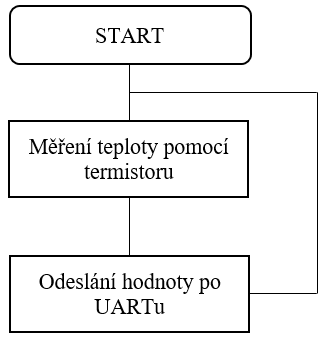
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Vyšší odborná škola a Střední průmyslová škola elektrotechnickáBožetěchova 3, OlomoucLaboratoře mikroprocesorové techniky | | | | | | | | | | | | | | |
| **SAMOSTATNÝ PROJEKT MIT** | | | | | | | | | | | | | | |
| Název úlohy | |  | | | | | | | | | | | Číslo úlohy |  |
| MĚŘENÍ TEPLOTY | | | | | | | | | | | | | MIT-02 | |
| Zadání   1. Měřte teplotu a posílejte jí na sběrnici UART | | | | | | | | | | | | | | |
| Poř. č. | Příjmení a jméno | | | |  | | | | | Třída | Skupina | | Školní rok |  |
| 15 | MICHALČÍK Ondřej | | | | | | | | | 4A | 2 | | 2021/22 | |
| Datum vypracování | | |  | Datum odevzdání | |  | Počet listů | |  | Klasifikace | | | | |
| 26. 4. 2022 | | | | 27. 4. 2022 | | | 4 | | |  | | | | | |
| Protokol obsahuje: | | | | | | | | Blokové schéma | | | | Program | | |
|  | | | | | | | | Vývojový diagram | | | | Závěr | | |
|  | | | | | | | | Schéma zapojení | | | |  | | |

# Zapojení:



Ke kitu STM8 je připojen termistor.

# Vývojový diagram:



# ZÁVĚR:

Výhodou programu je jeho jednoduchost. Ta umožňuje každému se v programu lehce vyznat a může tedy být použit jako takový příklad použití AD převodníku pro čtení napětí z termistoru a vysílání hodnot po sběrnici UART.

# Kód:

#include "stm8s.h"

#include "milis.h"

/\*#include "delay.h"\*/

#include "adc\_helper.h"

#include <stdio.h>

#include "stm8s\_adc2.h"

#include "uart1.h"

#define \_ISOC99\_SOURCE

#define \_GNU\_SOURCE

#define LED\_PORT GPIOC

#define LED\_PIN GPIO\_PIN\_5

#define LED\_REVERSE GPIO\_WriteReverse(LED\_PORT, LED\_PIN)

void setup(void)

{

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

GPIO\_Init(LED\_PORT, LED\_PIN, GPIO\_MODE\_OUT\_PP\_LOW\_SLOW);

ADC2\_SchmittTriggerConfig(ADC2\_SCHMITTTRIG\_CHANNEL4, DISABLE); //PB4

// 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\_4);

// rozběhneme AD převodník

ADC2\_Cmd(ENABLE);

// počkáme než se AD převodník rozběhne (~7us)

ADC2\_Startup\_Wait();

init\_milis();

init\_uart1();

}

int main(void)

{

uint32\_t time = 0;

uint16\_t ADCx;

uint16\_t napeti;

uint16\_t teplota;

setup();

while (1) {

if (milis() - time > 333) {

LED\_REVERSE;

time = milis();

ADCx = ADC\_get(ADC2\_CHANNEL\_4);

napeti = (uint32\_t)3300 \* ADCx / 1024;

teplota = ((uint32\_t)33000 \* ADCx - 4096000)/ 19968;

printf("U = %dmV\r\nTeplota = %d.%d°C\r\n", napeti, teplota/10, teplota%10);

}

}

}

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

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