

VŠB - TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

DIPLOMOVÁ PRÁCE

2017

Bc. Tomáš Szöcs

VŠB-TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
KATEDRA KYBERNETIKY A BIOMEDICÍNSKÉHO
INŽENÝRSTVÍ

Zařízení pro simulaci termoelektrických článků
Device for simulation of thermocouples

2017

Bc. Tomáš Szöcs

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání diplomové práce

Student: **Bc. Tomáš Szöcs**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2612T041 Řídicí a informační systémy
Téma: **Zařízení pro simulaci termoelektrických článků
Device for Simulation of Thermocouples**
Jazyk vypracování: čeština

Zásady pro vypracování:

1. Seznámení se s problematikou simulace termočlánků.
2. Návrh hardwaru zařízení pro simulaci termočlánku.
3. Realizace hardwaru zařízení pro simulaci termočlánku.
4. Implementace řídicího softwaru zařízení pro simulaci termočlánku.
5. Verifikace řešení a testování.
6. Zhodnocení výsledků závěrečné práce a závěr.

Seznam doporučené odborné literatury:


- [1] KREIDL, Marcel. *Měření teploty: senzory a měřicí obvody*. 1. vyd. Praha: BEN - technická literatura, 2005, 239 s. ISBN 80-7300-145-4.
- [2] MARTINEK, Radislav. *Senzory v průmyslové praxi*. 1. vyd. Praha: BEN - technická literatura, 2004. ISBN 80-7300-114-4.
- [3] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. 1. vyd. Ilustroval Viktorie VLACHOVÁ. Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-245-9.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **doc. Ing. Jan Židek, CSc.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017


doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

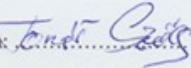



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 28.4.2017

Podpis studenta: 

Poděkování

Tímto bych chtěl poděkovat doc. Ing. Janu Žídkovi, CSc. za odborné vedení a rady během tvorby této práce.

Dále bych chtěl poděkovat Ing. Radimovi Hercikovi, PhD. a Ing. Petrovi Obadalovi za odborné konzultace, rady a cenné připomínky, především při tvorbě desky plošných spojů a zdrojového kódu.

V poslední řadě bych rád poděkoval rodině a přítelkyni za celkovou podporu při mém studiu.

Prohlášení zástupce spolupracující právnické osoby

Continental Automotive Czech Republic s.r.o.
Na Rovince 879
72000 Ostrava - Hrabová
Czech Republic


„Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.“

Z tohoto souhlasu jsou vyjmuty veškeré části bakalářské práce, které obsahují popis produktů, technického řešení, software, hardware a dále pak veškerého know-how a informací, které mohou být takto klasifikovány společností Continental Automotive Czech Republic s.r.o., které nejsou veřejně dostupnými informacemi.

S těmi částmi práce, musí být nakládáno jako s neveřejnou částí práce.



Continental Automotive Czech Republic, s.r.o.
Kopanská 1713, 744 01 Frenštát p.R.
IČ 62024922 DIČ CZ62024922 -26-


Ing. Radim Hercik, Ph.D.

V Ostravě 7.4.2017

Continental Automotive Czech Republic s.r.o.

Abstrakt

Tato práce pojednává o návrhu a vývoji zařízení, které umožňuje simulovat termoelektrické články. Simulátor je schopen simulovat až čtyři na sobě nezávislé termočlánky. Uživatel má možnost nastavit simulovanou teplotu termočlánku nebo simulované termoelektrické napětí v daném rozsahu. V případě simulování teploty termočlánku byl vytvořen polynom 6. řádu pro přepočítání simulované teploty na termoelektrické napětí. Pro zobrazení simulované veličiny je na simulátoru umístěn LCD displej. Simulátor je řízen 32 bitovým mikroprocesorem PIC32. Pro převod digitálního signálu na analogový, slouží DA převodníky. Digitální signál obsahuje hodnotu termoelektrického napětí. Mikroprocesor komunikuje s DA převodníky a LCD displejem pomocí linky SPI. Výstupní část simulátoru je galvanicky oddělena od řídicí a napájecí části obvodu. V teoretické části práce pojednává o termoelektrických jevech a termoelektrických člancích. V praktické části práce je popsán návrh DPS a následná implementace řídicího softwaru. Poslední část práce se zabývá testováním simulátoru.

Klíčová slova

termoelektrické jevy, termoelektrický článek, termoelektrické napětí, sériové periferní rozhraní (SPI), DA převodník, LCD displej, galvanický oddělovač, stabilizace napětí

Abstract

This thesis discusses a design and a development of the device, which allows simulate thermocouples. The simulator is able to simulate up to four independent thermocouples. The user has possibility to set the simulate temperature of thermocouple or the thermoelectric voltage in the range. In case of the simulating temperature of a thermocouple was formed sixth order polynomial, for conversion simulate temperature to the thermoelectric voltage. The simulator includes the LDC display to view simulating values. A simulator is controlled by 32 - bit microcontroller PIC32. Digital signal is converting to the analog signal, by the DA convertors. Digital signal contains value of thermoelectric voltage. Microcontroller communicate with DA convertors and LCD display by Serial Peripheral Interface (SPI). Output part of simulator is galvanically separated from the control and power part of the circuit. Theoretical part of thesis is describing thermoelectric phenomenoms and thermocouples. Practical part of thesis is describing the design and the development of PCB and following implementation of the control software. The last part of thesis is about testing the simulator.

Key words

thermoelectric phenomenon, thermocouple, thermoelectric voltage, serial peripheral interface (SPI), DA converter (DAC), LCD display, galvanic isolator, voltage stabilization

Obsah

Seznam obrázků	10
Seznam tabulek	12
1 Úvod.....	13
2 Teoretický rozbor	15
2.1 Termoelektrické jevy	15
2.2 Kontaktní napětí	15
2.3 Seebeckův jev	16
2.4 Peltierův jev	19
2.5 Thomsonův jev.....	20
2.6 Termoelektrické články.....	22
2.7 Termoelektrické materiály	24
2.8 Konstrukce termoelektrických článků.....	25
3 Návrh desky plošných spojů	26
3.1 Napěťové úrovně napájecí části	26
3.1.1 Stabilizace napětí na 5V	26
3.1.2 Stabilizace napětí na 3,3V	27
3.2 Procesor.....	28
3.2.1 Serial Peripheral Interface (SPI)	28
3.3 LCD display	30
3.4 Izolátory	31
3.5 Napěťová reference.....	32
3.6 DA převodník.....	33
3.7 Operační zesilovač	35
3.8 Napěťový dělič.....	36
3.9 Schéma zapojení	36
3.10 Rozmístění součástek na DPS	39
3.11 Osazení DPS	41
4 Implementace softwaru	43
4.1 Softwarová architektura	43
4.1.1 Vývojový diagram řídicího softwaru	44
4.2 Konfigurace mikroprocesoru	46
4.3 Použité knihovny.....	47
4.3.1 Knihovna SPI	48
4.3.2 Knihovna LCD	49
4.4 Řídicí software	52
4.4.1 Enkodéry	52
4.4.2 Funkce TC_VOLTAGE.....	55
4.4.3 Funkce TC_TEMP	58
4.4.4 Přepínání zobrazení simulované veličiny	62
5 Testování a verifikace	64

6	Závěr	68
7	Seznam použité literatury.....	70
	Seznam příloh	72

Seznam obrázků

Obr. 2.1 Objemový Seebeckův jev pro polovodič typu N, $T_A > T_B$ (neuzavřený obvod; uzavřený obvod)	18
Obr. 2.2 Objemový Seebeckův jev pro polovodič typu P, $T_A > T_B$ (neuzavřený obvod; uzavřený obvod)	18
Obr. 2.3 Schéma pro demonstraci Peltierova jevu.....	19
Obr. 2.4 Peltieruv jev - proud I_Z má stejnou orientaci než I_S	20
Obr. 2.5 Peltieruv jev - proud I_Z má opačnou orientaci než I_S	20
Obr. 2.6 Thomsonův jev	21
Obr. 2.7 Obecné schéma termoelektrického článku	23
Obr. 2.8 Typy termočlánků [10].....	25
Obr. 2.9 Konstrukční schéma termočlánku [11]	25
Obr. 3.1 DC/DC konvertor.....	27
Obr. 3.2 Schéma zapojení regulátoru LM2937.....	28
Obr. 3.3 SPI master a slave.....	29
Obr. 3.4 Zobrazení napětí na displeji.....	30
Obr. 3.5 Zobrazení teploty na displeji.....	30
Obr. 3.6 Schéma zapojení LCD displeje	31
Obr. 3.7 Schéma zapojení izolátoru	31
Obr. 3.8 Vnitřní struktura izolátoru	32
Obr. 3.9 Schéma zapojení reference	33
Obr. 3.10 Schéma zapojení DA převodníku	33
Obr. 3.11 Schéma zapojení operačního zesilovače	35
Obr. 3.12 Schéma zapojení napěťového děliče	36
Obr. 3.13 Napájecí část obvodu	37
Obr. 3.14 Použití funkce SHOW.....	38
Obr. 3.15 Návrh DPS - varianta A; varianta B.....	39
Obr. 3.16 Konečná podoba DPS.....	40
Obr. 3.17 Osvětlová maska	41
Obr. 3.18 Plně osazená DPS.....	41
Obr. 4.1 Blokový diagram vrstev.....	44
Obr. 4.2 Vývojový diagram	45
Obr. 4.3 Bitová konfigurace mikroprocesoru.....	47
Obr. 4.4 Inicializace SPI linky	48
Obr. 4.5 Zápis dat přes SPI linku	49
Obr. 4.6 Vyhledávací tabulka (LookUp Table)	50
Obr. 4.7 Funkce pro výběr řádku na displeji	50
Obr. 4.8 Zápis textového řetězce	51
Obr. 4.9 Vytvoření bytu pro zápis přes SPI.....	51
Obr. 4.10 Zápis dat na displej přes SPI	52
Obr. 4.11 Řídící funkce pro enkodéry	54

Obr. 4.12 Převod ASCII znaků na čísla - TC_VOLTAGE.....	56
Obr. 4.13 Přepočítání napětí na datové slovo - TC_VOLTAGE	57
Obr. 4.14 Zobrazení simulovaného napětí na displeji	57
Obr. 4.15 Převod ASCII znaků na čísla - TC_TEMP.....	58
Obr. 4.16 Bodový graf s regresní křivkou	60
Obr. 4.17 Přepočítání napětí na datové slovo - TC_TEMP	61
Obr. 4.18 Zobrazení simulované teploty na displeji.....	61
Obr. 4.19 Řídící funkce pro přepínání zobrazení simulované veličiny	63
Obr. 5.1 Použitý měřicí přístroj [14].....	64
Obr. 5.2 Grafické porovnání výsledků měření	67
Obr. 5.3 Simulátor během testování.....	67

Seznam tabulek

Tab. 2.1 <i>Výstupní napětí kovů</i> [3]	16
Tab. 2.2 <i>Seebeckovy koeficienty</i> [5]	18
Tab. 2.3 <i>Rozsahy teplot nejčastěji používaných termočlánků</i> [6]	24
Tab. 4.1 <i>Referenční tabulka</i>	59
Tab. 5.1 <i>Naměřené hodnoty a jejich odchylky – simulace napětí</i>	65
Tab. 5.2 <i>Naměřené hodnoty a jejich odchylky – simulace teploty</i>	66

1 Úvod

Cílem této diplomové práce je navrhnout a vytvořit zařízení, které má sloužit pro simulaci termoelektrického článku. Zadavatelem této práce je firma Continental Automotive Czech Republic s. r. o. Pro firmu Continental je vývoj tohoto simulátoru důležitý především pro další vývoj senzorů. Simulátor vytvoří vývojářům firmy Continental lepší podmínky hlavně při testování těchto senzorů a systémů.

Podle zadání zadavatele pracuje navržené zařízení samostatně, bez nutnosti připojení k počítači a součinnosti se SW aplikací na PC. Pro plnohodnotný provoz zařízení tedy není potřeba žádných externích periférií. Napájecí napětí navrženého simulátoru je 12V DC. Činnost simulátoru je řízena mikroprocesorem PIC32. Simulátor dokáže simulovat až 4 na sobě nezávislé termočlánky. Součástí simulátoru je také LCD displej, který zobrazuje termoelektrické napětí na výstupu daného kanálu simulátoru a simulovanou teplotu jednotlivých termočlánků. Pomocí inkrementálních aktuátorů může uživatel nastavit v daném rozsahu teplotu nebo termoelektrické napětí pro každý simulovaný termočlánek (kanál) zvlášť. Jelikož termočlánky umožňují měření ve velkém teplotním rozsahu, jsou na zařízení umístěna tlačítka pro změnu velikosti inkrementu. Základní inkrementace je po jedné desetině stupně Celsia nebo milivoltu. Změna velikosti inkrementu umožňuje inkrementovat napětí nebo teplotu po jednotkách nebo desítkách stupňů Celsia nebo milivoltů.

Uživateli je dále poskytnut výběr zobrazení teploty nebo termoelektrického napětí na displeji simulátoru. Nastavení zobrazení je pro všechny 4 zobrazované kanály stejné, není možné zobrazovat pro první dva kanály teplotu a pro další dva kanály termoelektrické napětí apod. Dále má uživatel možnost nezávisle měnit hodnotu teploty nebo termoelektrického napětí jednotlivých kanálů podle aktuálně zobrazované veličiny.

Simulátor obsahuje také mini USB port. Tento port slouží k ovládní simulátoru prostřednictvím aplikace pro sériovou komunikaci běžící na PC jako je například *RealTerm*. Pomocí jednoduchých příkazů lze simulátor inicializovat, nastavit na daném kanále přesnou hodnotu termoelektrického napětí nebo teploty.

Rozsah výstupního napětí je dán parametry výstupních DA převodníků. Jelikož se termoelektrické napětí pohybuje řádově v milivoltech, je zapotřebí vícebitový převodník s co největším počtem kroků. Součástí této práce je rovněž test funkčnosti jednotlivých kanálů a měření dat, ze kterých je sestavena charakteristika zobrazující závislost termoelektrického napětí na simulované teplotě. Následně je tato naměřená veličina porovnána s veličinou, která je vytvořena z hodnot, jež jsou udávány výrobcem v katalogovém listu daného termočlánku.

V teoretické části práce jsou popsány termoelektrické jevy z fyzikálního hlediska a rovněž jejich využití. Tato část práce dále popisuje principy měření teploty prostřednictvím termoelektrických článků. Dále jsou zde popsány jednotlivé typy termoelektrických článků a jejich dělení podle konstrukce, využití v praxi a v neposlední řadě termoelektrické materiály, ze kterých

se dané články vyrábí. V neposlední řadě jsou v této části obecně popsány komponenty, které jsou použity na desce plošných spojů celého zařízení.

V úvodu praktické části je popsán celkový návrh desky plošných spojů. Úvodní kapitola se přímo zabývá návrhem obvodového schématu. Další kapitola pak popisuje rozmístění součástek na desce plošných spojů a vytvoření vodivých cest. V následující kapitole je popsán vývoj softwaru pro jednotlivé periferie simulátoru a také vývoj celkového řídicího softwaru. Řídicí software byl vyvinut ve vývojovém prostředí MPLAB, společnosti Microchip Technology Inc. Praktická část práce popisuje též konečné testování jednotlivých kanálů simulátoru, včetně porovnání naměřené veličiny s veličinou sestavenou z dat katalogového listu simulovaného termočlánku.

2 Teoretický rozbor

2.1 Termoelektrické jevy

Termoelektrické vlastnosti látek v kterémkoli skupenství mají stejný, v konkrétních aplikacích i vyšší význam, nežli vlastnosti elektrické, mechanické, magnetické a jiné. Termoelektrické jevy mají především význam z hlediska transformace tepelné energie na energii elektrickou a naopak. Vzájemná podmíněnost teplotních a potenciálových gradientů způsobuje, že za určitých okolností gradient teploty generuje gradient potenciálu a naopak gradient potenciálu vytváří teplotní spád. [1]

Termoelektrické jevy můžeme rozdělit do dvou skupin, a sice na jevy, při kterých z rozdílu teplot vzniká termoelektrické (přesněji termoelektromotorické) napětí – Seebeckův a Benedicksův jev. Do druhé skupiny patří jevy, při kterých je průtokem stejnosměrného elektrického proudu uvolňováno a naopak pohlcováno teplo – Peltierův a Thomsonův jev. Tyto jevy nebyly ve své době příliš významné. Vzhledem k tomu, že účinnost termoelektrických jevů jako zdrojů elektrické energie je velmi nízká, byly tyto jevy zastíněny objevem elektromagnetické indukce Michaela Faradaye (1791 - 1867). Své místo v elektrotechnice si našly teprve až s příchodem polovodičů. [2]

2.2 Kontaktní napětí

V atomu každého kovu najdeme minimálně jeden valenční elektron, který je k jádru vázán jen velmi slabými silami, a proto může být snadno uvolněn. Volně se pohybující elektrony pak tvoří tzv. elektronový plyn. Aby tyto elektrony mohly být uvolněny mimo kov, je zapotřebí jim dodat energii (v našem případě tepelnou) tak, aby jeho kinetická energie byla větší než síla, která ho drží uvnitř kovu. Síla, která drží elektron uvnitř kovu, se nazývá Coulombova síla. Hovoříme tedy o vykonané práci A . Pro kterou platí:

$$A = e \cdot U \tag{2.1}$$

kde: e je velikost náboje elektronu [$1,6 \cdot 10^{-19}C$]

U je velikost výstupního napětí

A je velikost výstupní práce [eV]

Hodnoty výstupních napětí některých kovů jsou uvedeny v tabulce. Pokud tedy spojíme dva kovy s různými výstupními napětími (vlastními potenciály), vznikne na jejich spojích kontaktní napětí, které odpovídá rozdílu těchto výstupních napětí (vlastních potenciálů).

Pro kontaktní napětí platí vztah:

$$U_{AB} = U_B - U_A \quad (2.2)$$

kde: U_{AB} je kontaktní napětí kovů A a B

U_B je výstupní napětí kovu A

U_A je výstupní napětí kovu B

Tab. 2. 1 Výstupní napětí kovů [3]

Kov	U [V]	Kov	U [V]
Na - sodík	2,33	Sn - cín	4,51
K - draslík	2,26	Pb - olovo	4,15
Cs - cesium	1,93	Sb - antimon	4,02
Cu - měď	4,29	Bi - bismut	4,62
Ag - stříbro	4,73	Co - kobalt	3,72 - 4,25
Au - zlato	4,76	Fe - železo	4,75 - 4,77
Mg - hořčík	3,69	Ni - nikl	4,98 - 5,03
Zn - zinek	4,25	Rh - rhodium	4,75
Al - hliník	4,25	Pt - platina	5,44 - 6,37
Ge - germanium	4,55	Se - selen	4,89

Tento jev, ale nelze využít jako zdroj elektrického proudu, protože v uzavřeném obvodu se kontaktní napětí navzájem vyruší, ovšem pouze za předpokladu konstantní teploty na všech kovech v daném obvodu. [3]

2.3 Seebeckův jev

Seebeckův jev je přeměna teplotních rozdílů přímo na elektrické napětí. Tento jev byl náhodně objeven v roce 1821 německým fyzikem Thomasem Johannem Seebeckem, který zjistil, že existuje elektrické napětí mezi dvěma konci kovové tyče, pokud mezi těmito konci existuje teplotní gradient ΔT .

Zjistil také, že kompasová střílka je odkloněna, pokud je vytvořena uzavřená smyčka ze dvou různých kovů s teplotním rozdílem mezi spoji. Je tomu tak z důvodu rozdílné odezvy kovů na teplotní rozdíly, což vytváří proudovou smyčku, která způsobuje vznik magnetického pole.

Seebeckův jev je vznik napětí, který nastává při teplotních rozdílech mezi dvěma odlišnými kovy nebo polovodiči. To způsobuje nepřetržité proudění elektronů, pokud vodiče vytvoří uzavřený obvod. Vzniklé napětí je v řádu několika mikrovoltů na stupeň Celsia teplotního rozdílu.

Výsledné termoelektrické napětí lze vypočítat dle tohoto vztahu:

$$U_{AB} = \int_{T_1}^{T_2} (\alpha_B(T) - \alpha_A(T)) dT \quad (2.3)$$

kde: U_{AB} je termoelektrické napětí mezi kovy A a B

α_A, α_B jsou Seebeckovy koeficienty

T je teplota spojů

Seebeckovy koeficienty jsou nelineární a závisejí na teplotě vodičů, použitém materiálu a jeho molekulární struktuře. Pokud jsou Seebeckovy koeficienty v daném rozsahu teplot přibližně konstantní, může být výše uvedená rovnice (3.1) linearizována: [1, 4]

$$U_{AB} = (\alpha_A - \alpha_B) \cdot (T_B - T_A) + 0,5(\beta_A - \beta_B) \cdot (T_B - T_A)^2 \quad (2.4)$$

kde: U_{AB} je termoelektrické napětí mezi kovy A a B

$\alpha_A, \alpha_B, \beta_A, \beta_B$ jsou Seebeckovy konstanty

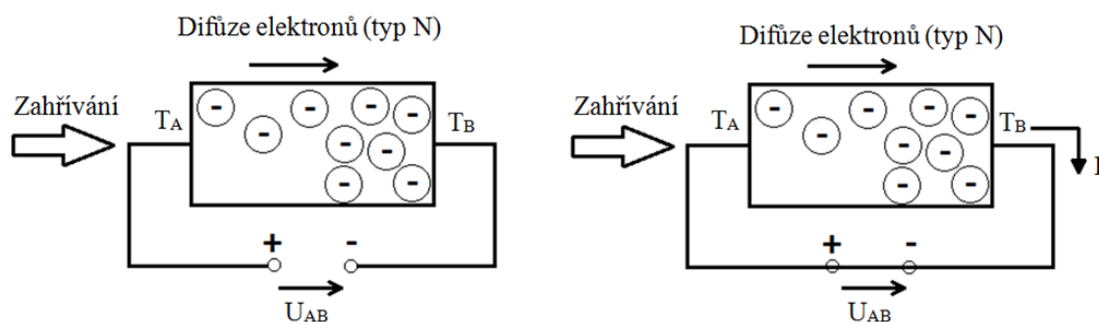
T_A, T_B jsou teploty spojů [K]

Seebeckův jev lze rozdělit na dva typy, a sice kontaktní a objemový. Kontaktní Seebeckův jev vzniká důsledkem kontaktního potenciálu, který na rozhraní dvou látek vždy vzniká. Jestliže jsou však oba kontakty na stejné teplotě, je celkové elektromotorické napětí v obvodu nulové. Kontaktní termoelektrický jev tedy závisí na změně Fermiho energie s teplotou, a je prakticky jedinou příčinou tohoto jevu v kovech. Poskytuje hodnoty napětí kolem jednotek až desítek $\mu V/K$ (Tab. 2. 2). [1]

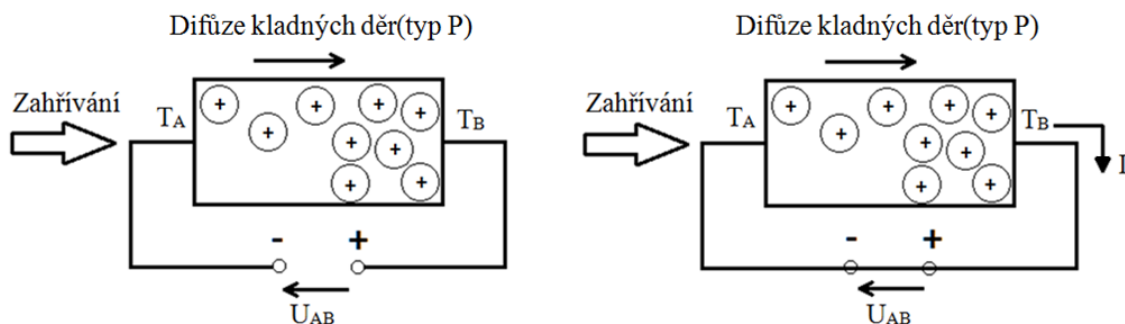
Tab. 2. 2 Seebeckovy koeficienty [5]

Kov	α [$\mu\text{V/K}$]	β [$\mu\text{V/K}^2$]
Antimon	35,6	0,145
Bizmut	-74,4	0,032
Konstantan	-38,1	-0,0888
Měď	2,71	0,0079
Nikl	-19,1	-3,02
Platina	-3,03	-3,25
Železo	16,7	-0,0297

Objemový Seebeckův jev vzniká důsledkem difúze nosičů náboje a je hlavním zdrojem termoelektrického napětí v polovodičích. V místech s vyšší teplotou, je vyšší koncentrace nosičů náboje, takže začne jejich difúze do míst s nižší teplotou, kde se tyto náboje začnou hromadit. Jestliže uzavřeme tento obvod, dojde k pohybu elektronů přes přechod, tedy obvodem začne protékat elektrický proud. Tím se látka elektricky nabíjí - v případě N typu vodivosti (Obr. 2. 1) je teplý konec nabitý kladně, studený záporně, v případě P typu vodivosti (Obr. 2. 2) opačně. [5]



Obr. 2. 1 Objemový Seebeckův jev pro polovodič typu N, $T_A > T_B$ (neuzavřený obvod; uzavřený obvod)



Obr. 2. 2 Objemový Seebeckův jev pro polovodič typu P, $T_A > T_B$ (neuzavřený obvod; uzavřený obvod)

2.4 Peltierův jev

Peltierův jev spočívá v uvolňování nebo pohlcování tepla v oblasti styku dvou vodičů z různých materiálů, kterými prochází elektrický proud. Vzniká v důsledku změny entropie nosičů náboje při jejich průchodu přechodem. Jedná se tedy o opak Seebeckova jevu. Zde dochází k uvolňování nebo pohlcování tepla podle směru procházejícího elektrického proudu. Množství tohoto tepla je přímo úměrné velikosti proudu a platí vztah:

$$Q = \Pi_{AB} \cdot I \cdot t \quad (2.5)$$

kde: Π_{AB} je Peltierův koeficient

I je elektrický proud

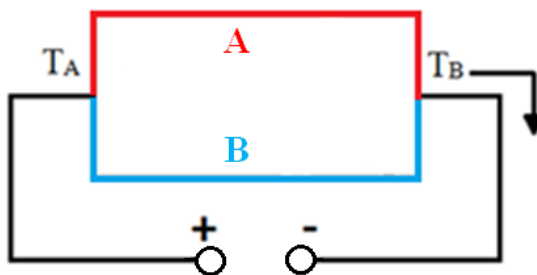
t je doba průchodu elektrického proudu

Pro Peltierův koeficient platí následující vztah:

$$\Pi_{AB} = \Pi_A - \Pi_B \quad (2.6)$$

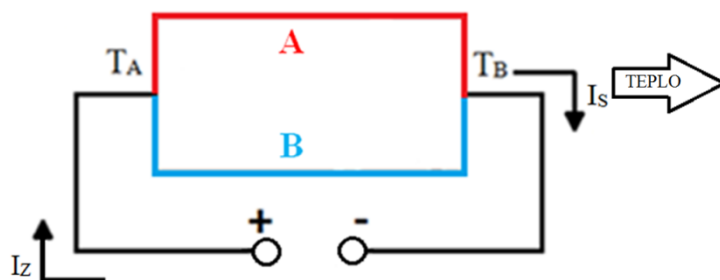
kde: Π_{AB} je Peltierův koeficient

Π_A, Π_B jsou Peltierovi koeficienty jednotlivých materiálů

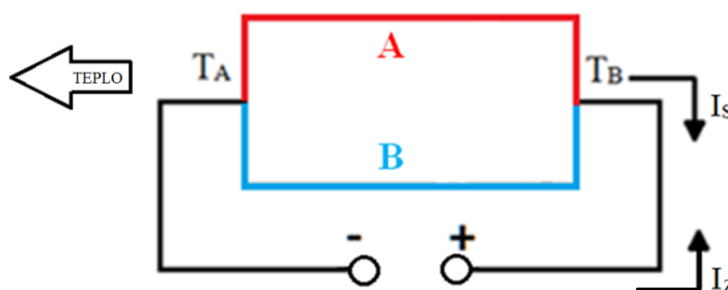


Obr. 2. 3 Schéma pro demonstraci Peltierova jevu

Připojíme-li nyní tento obvod k vnějšímu zdroji stejnosměrného napětí, aniž bychom vytvořili rozdíl teplot, nastane Peltierův jev. Obvodem začne procházet elektrický proud, který vytvoří rozdíl teplot ve spojích. Během Peltierova jevu se začne jeden konec ochlazovat a druhý oteplovat. Ochlazování a oteplování daného konce závisí na směru protékajícího proudu. Názorně je to zachyceno na obrázcích níže. (Obr. 2. 4, Obr. 2. 5) [7, 8]:



Obr. 2. 4 Peltieruv jev - proud I_Z má stejnou orientaci než I_S



Obr. 2. 5 Peltieruv jev - proud I_Z má opačnou orientaci než I_S

2.5 Thomsonův jev

V roce 1857 William Thomson (lord Kelvin) teoreticky popsal jev, nazvaný po něm Thomsonův jev, který sice Seebeck již v minulosti experimentálně zjistil, ale nevěnoval mu pozornost. Termoelektrické napětí vzniká i v každém osamoceném jediném homogenním vodiči, jsou-li na něm místa s rozdílem teplot. V místech s rozdílnou teplotou se vlastnosti vodiče nepatrně liší, což je příčinou vzniku dalšího termoelektrického napětí, nazývaného napětím Thomsonovým.

Princip Thomsonova jevu spočívá v tom, že máme homogenní kovovou tyč délky l m, kterou na jednom konci zahříváme na teplotu T_A a druhý konec udržujeme při stálé teplotě T_B . Rozdílem teplot vznikne tzv. teplotní spád (gradient), který je dán vztahem:

$$\text{grad}T = \frac{\Delta T}{l} \text{ [K} \cdot \text{m}^{-1}] \quad (2.7.)$$

kde: ΔT je rozdíl teplot na jednom a druhém konci tyče [K]

l je délka vodivé tyče [m]

Mezi konci vodiče vznikne nepatrné termoelektrické napětí U (Obr. 2. 6). Tomuto napětí odpovídá elektrostatické pole popsané vnitřní elektrickou intenzitou \vec{E}_i .

S Thomsonovým vztahem úzce souvisí Joulovo teplo. Pro výpočet tepla sálajícího z kovové tyče platí vztah:

$$Q = RI^2 - \mu I \frac{\Delta T}{l} \quad (2.8.)$$

kde: Q je teplo sálající z vodivé tyče

R je elektrický odpor vodivé tyče [Ω]

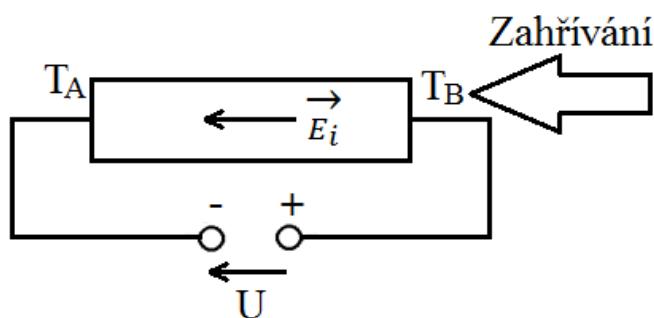
I je elektrický proud [A]

μ je Thomsonův koeficient [$V \cdot K^{-1}$]

ΔT je rozdíl teplot na jednom a druhém konci tyče [K]

l je délka vodivé tyče [m]

První člen vztahu odpovídá již zmiňovanému, Jouleovu teplu, které se uvolňuje do okolí vodiče, prochází-li jím elektrický proud. Druhý člen je Thomsonovo teplo, které vzniká díky teplotnímu gradientu ve vodiči. Znaménko Thomsonova tepla se mění v závislosti na směru elektrického proudu, který vodičem protéká.



Obr. 2. 6 Thomsonův jev

Thomsonův jev je velice podobný Seebeckovu jevu, ale rozdíl je v tom, že Thomsonův jev vzniká pouze na jednom vodiči, přičemž u Seebeckova jevu je nutné spojení dvou vodičů. Z tohoto důvodu je při Thomsonově jevu termoelektrické napětí nižší, než u Seebeckova jevu.

Kolem roku 1921 prezentoval Carl Benedicks (1875 - 1958) objev zatím posledního známého termoelektrického jevu – Benedicksova jevu. Hodnota termoelektrického napětí je zde určena

pouze velikostí teplotního gradientu. Tento jev pojednává o tom, že i při shodné teplotě konců vodiče je mezi nimi elektrické napětí, za předpokladu existujícího teplotního gradientu v některé jeho části. Toto napětí je u kovů téměř neměřitelné, a proto byl Benedicksův jev poprvé prokázán až u polovodičů, konkrétně na germaniu. [3]

2.6 Termoelektrické články

Termočlánek je elementární prvek tvořený spojením dvou různých materiálů, který je využíván výhradně jako senzor pro snímání teploty. Využívá principu Seebeckova jevu, tedy přímé přeměny tepelné energie na elektrickou. V elektrotechnice je nejrozšířenějším snímačem teploty především pro svou jednoduchost a široký teplotní rozsah. Přesnost dnešních termočlánků je při měření teplot mnohdy až $\pm 0,1$ °C. Toto je výsledkem dlouholetého vývoje výrobních postupů a vývoje složení materiálů. Termočlánky jsou nejčastěji vyráběny ze dvou kovů. Současná konstrukce termočlánků jim zajišťuje velké množství výhod, kterými jsou:

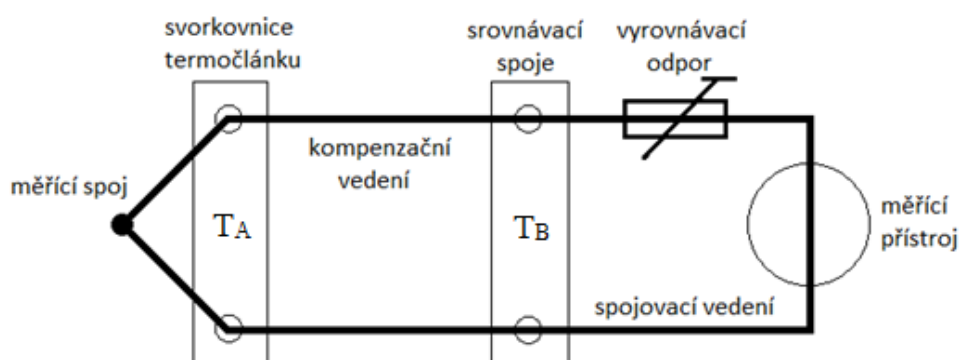
- nízká teplotní setrvačnost – téměř okamžitá reakce na změny teplot
- nízká tepelná kapacita – k jeho ohřátí stačí pouze malé množství tepla
- vysoká citlivost – tím také poměrně velká přesnost
- možnost měření teplot i na velmi dlouhou vzdálenost od měřeného místa
- široký teplotní rozsah (přibližně -250 °C až 1600 °C)

Termoelektrické články můžeme z hlediska využití a použitých materiálů rozdělit do dvou hlavních kategorií. Termoelektrické články sloužící pro měření teplot a termoelektrické články pro přeměnu tepelné energie na energii elektrickou a naopak. Termoelektrické články pro snímání teplot jsou vyráběny především z kovů, naopak termoelektrické články pro přeměnu energie jsou vyráběny výhradně z polovodičů. Základní dělení termoelektrických článků:

- Termoelektrické články pro měření teplot
 - Typ S (Pt - RhPt)
 - Typ R (Pt - RhPt)
 - Typ K (NiCr - Ni)
 - Typ J (FeCu - Ni)
 - Typ M (Ni - Mo)
 - Typ T (Ni - Cu)
 - Typ L (FeCu - Ni)
 - Typ E (Ni - CuCr)
 - Typ N (NiCr - Si)
 - Typ B (Pt - RhPt)
 - Typ A (W - ReW)
- Termoelektrické články pro přeměnu energie
 - Peltierovy články (TEC)
 - Termoelektrické generátory (TEG)

Na obrázku níže (Obr. 2. 7) je znázorněn obecný termoelektrický článek včetně označení jednotlivých částí termočlánku. Při popisu z levé strany směrem doprava, je první na řadě měřicí

spoj. Měřicí spoj je část termočlánku, kde jsou materiály pevně a vodivě spojeny. Měřicí spoj umístíme do přímého nebo velmi blízkého kontaktu s měřeným prostředím o teplotě T_A . Spolu s dostatečně dlouhou částí obou větví tvoří termočlánek. Prodlužovací vedení termočlánku je vyrobeno ze stejného materiálu jako měřicí spoj termočlánku, nejčastěji se používají kovy (Fe, Cu, Al). To znamená, že například termočlánek FeCu bude mít jednu větev prodlužovacího vedení vyrobenou ze železa a druhou z mědi. Kompenzační vedení, které slouží k prodloužení termočláneků je vyrobeno převážně z vzácných kovů (Pt, Au, Rh, Ag, Pd). Je-li kompenzační vedení vyrobeno z kovů jako jsou železo, měď atd. musí splňovat stejné nebo alespoň přibližně stejné termoelektrické vlastnosti, a v daném teplotním rozsahu i charakteristiku, jako tyto drahé kovy. Prodlužovací vedení je ukončeno v místě, kde již nemůže být ovlivněno zdrojem tepla z měřeného místa. Zde jsou umístěny srovnávací spoje obou větví, ve kterých je vztažná teplota T_B . Teplota T_B by měla být konstantní. Od srovnávacích spojů je dále vedeno spojovací vedení, které může být tvořeno kterýmkoli běžně používaným vodičem (Cu, Al), jelikož třetí kov vložený mezi srovnávací spoj nemá na funkci termočlánku žádný vliv. V obvodu plní funkci pouhého vodivého propojení termočlánku s citlivým měřicím přístrojem. [2]



Obr. 2. 7 Obecné schéma termoelektrického článku

V dnešní době jsou již normalizovány základní páry termoelektrických materiálů pro výrobu termočláneků, které se liší svými vlastnostmi (odolnost proti korozi, nelinearita napětí při vysokých teplotách, odolnost proti chemickým vlivům a ionizačnímu záření). Dříve se druhy termočláneků odlišovaly pomocí barev, dnes jsou termočláneků značeny velkými písmeny abecedy. Typy termočláneků a jejich teplotní rozsahy jsou uvedeny v tabulce. (Tab. 2. 3) V České republice se nejčastěji používají levnější typy J a K. Typ J (železo-konstantan) a K (chromel-alumel) lze použít v oxidační a inertní atmosféře. Termočlánek typu J lze navíc použít v redukční atmosféře a vakuu. [2]

Tab. 2. 3 Rozsahy teplot nejčastěji používaných termočlánků [6]

materiál termočlánku	označení	rozsah teplot při dlouhodobém použití (°C)	maximální teplota při krátkodobém použití (°C)
měď / kopel	M	- 200 ... + 100	+ 100
měď / měď – nikl	T	- 200 ... + 400	+ 400
chromel / kopel	L	- 200 ... + 600	+ 800
nikl – chrom / měď – nikl	E	- 100 ... + 700	+ 900
železo / měď – nikl	J	- 200 ... + 700	+ 900
nikl – chrom – křemík / nikl – křemík	N	- 200 ... + 1000	+ 1200
nikl – chrom / nikl – hliník	K	- 200 ... + 1000	+ 1300
platina - 10 % rhodium / platina	S	0 ... + 1300	+ 1600
platina - 13 % rhodium / platina	R	0 ... + 1300	+ 1600
platina - 30 % rhodium / platina – 6 % rhodium	B	+ 300 ... + 1600	+ 1800
wolfram – 5 % rhenium / wolfram – 20 % rhenium	A	0 ... + 2200	+ 2500

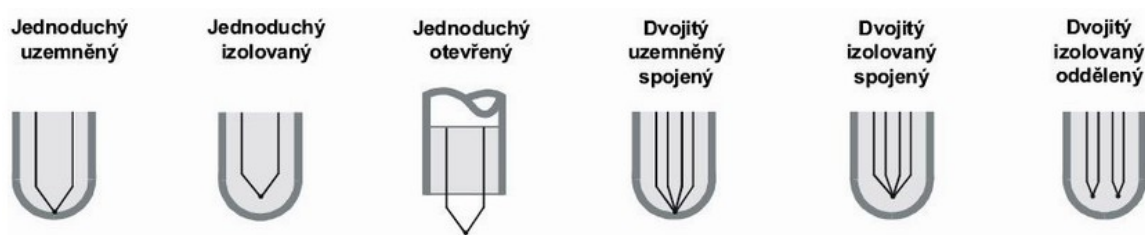
2.7 Termoelektrické materiály

Pro termoelektrické materiály je důležité, aby splňovaly základní požadavky. Hlavním požadavkem je, aby závislost termoelektrického napětí na teplotě měla co nejvíce lineární průběh. Termoelektrický materiál musí být odolný proti chemickým, mechanickým a korozním vlivům. Výstupní termoelektrické napětí by mělo být co největší. Čím je hodnota napětí menší, tím citlivější musí být měřicí přístroj, a tím méně může docházet k nepřesnostem a zkreslení. U snímačů z křehkých kovů (vizmut, antimon apod.) je nutno tyto kovy na sebe ve vakuu nastříkat nebo spékat.

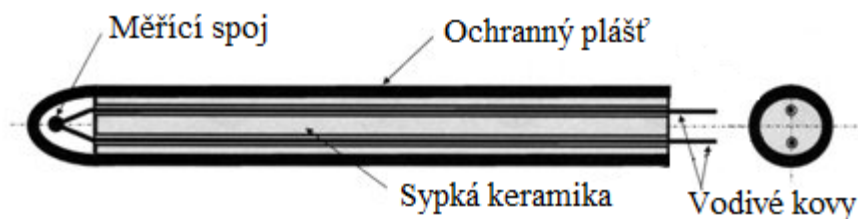
Při výběru materiálu je třeba myslet na požadovaný rozsah teplot a požadované přesnosti měření. Důležitá je také časová stálost, popř. střední doba životnosti snímače. Charakteristika termočlánku by měla být s časem neproměnná. Tato podmínka je však těžko splnitelná, jelikož dochází k rekrystalizaci v místě spoje. Snímače se pak musí obnovovat a občas přecejchovat. Byly sestaveny dvojice materiálů, které dosahují nejlepších vlastností a používají se ke konstrukci termoelektrických snímačů. V české státní normě jsou cejchovní řady termoelektrických článků Fe - ko, ch-a, ch-k (chromal-kopel) a PtRh - Pt. Jsou v ní udány hodnoty napětí odstupňované po 10°C po vztažnou teplotu srovnávacího spoje 20°C. Pokud je teplota srovnávacího spoje jiná (např. 0 nebo 50°C), pak musíme hodnoty napětí přepočítat. U termoelektrických snímačů je třeba počítat i s tím, že snímače téhož druhu mohou mít různý původ, a tedy i různé cejchovní řady, takže jejich charakteristiky jsou poněkud rozdílné. Rozdíl je dán složením použitých materiálů. [9]

2.8 Konstrukce termoelektrických článků

Termoelektrické články můžeme z pohledu konstrukce rozdělit na dva základní typy, a sice na plášťované a neplášťované. Použití jednotlivých článků je určeno převážně prostředím, ve kterém budou termočlánky umístěny pro měření. V případě měření vysokých teplot nebo v chemicky agresivním prostředí se používají plášťované termočlánky. Vodiče termočlánku jsou uvnitř tohoto pláště elektricky odděleny sypkou keramikou. V některých oborech, jako je například jaderná energetika, se z bezpečnostních důvodů počet vodičů v termočláncích zdvojuje, aby bylo zajištěno měření i při přerušení některého z drátků, jedná se o tzv. dvojitě termočlánky. Na obrázcích níže můžeme vidět rozdělení termočlánků a jejich vnitřní strukturu.



Obr. 2. 8 Typy termočlánků [10]



Obr. 2. 9 Konstrukční schéma termočlánku [11]

3 Návrh desky plošných spojů

V následujících kapitolách je popsána praktická část práce, která obsahuje návrh obvodového schématu a návrh desky plošných spojů. Osazená a oživená deska plošných spojů pak tvoří výsledný simulátor. Pro návrh desky plošných spojů existuje v dnešní době celá řada softwarových nástrojů. Jelikož jsem měl zkušenosti s návrhem DPS ve vývojovém prostředí *Eagle*, byla pro mne volba jednoduchá. Celé deska je tedy navržena ve vývojovém prostředí *Eagle*.

Nejdůležitější část zařízení je právě ta výstupní. V této části dochází ke konečné úpravě signálu, tedy k převedení z digitální podoby na analogovou. Analogový signál, který je převedený pomocí DA převodníku, je následně upraven napěťovým děličem na požadovanou velikost, v tomto případě se jedná o milivolty.

3.1 Napěťové úrovně napájecí části

Aby zařízení mohlo vůbec fungovat, je potřeba elektrická energie. Na desce plošných spojů jsou umístěny řádově desítky součástek, pro které je zapotřebí zajistit několik napěťových úrovní napájecího napětí. Celé zařízení je napájeno z elektrické sítě, která poskytuje střídavé napětím o velikosti jmenovité efektivní hodnoty 230V. Základní vstupní napětí potřebné pro napájení simulátoru je stejnosměrné, o velikosti 12V. Střídavé napětí z elektrické sítě je tedy potřeba usměrnit a upravit jeho velikost. K úpravě a usměrnění napětí slouží stabilizovaný zdroj napětí, jehož vstupní napětí je střídavé 230V a výstupní napětí stejnosměrné 12V.

Na vstup desky plošných spojů je tedy přivedeno stejnosměrné napětí 12V. Součástky umístěné na desce vyžadují dvě různé napěťové úrovně. Vstupní napětí 12V je potřeba upravit na úrovně 5V a 3,3V.

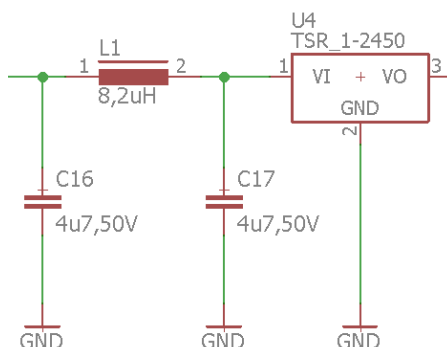
3.1.1 Stabilizace napětí na 5V

Jelikož upravujeme stejnosměrné napětí na stejnosměrné, tedy upravujeme pouze hodnotu, není potřeba toto napětí usměrňovat. Takovou úpravu napětí lze provést více způsoby.

Jednou z možností je použití napěťového děliče. Tato možnost vyžaduje použití více součástek, konkrétně rezistorů, a navíc není zdaleka tak přesná. Hlavní nevýhodou napěťového děliče je změna výstupního napětí při jeho zatížení. Dále také závisí na přesnosti rezistorů. Přesnost rezistorů udává tolerance, která je uváděna v procentech. Hodnoty rezistorů je nutné vypočítat dle příslušných vztahů. Vypočítané hodnoty ale nemusí odpovídat hodnotě rezistoru v odporové řadě pro danou přesnost, a je tak nutné vypočítané hodnoty rezistoru získat kombinací více odporů. Z těchto důvodů jsem onu metodu pro úpravu napětí nepoužil.

Další z možností je použití tzv. DC/DC konvertoru. Jedná se o aktivní součástku, která upravuje vstupní napětí na požadované výstupní napětí. Vstupní napětí bývá v určitém rozsahu a výstupní napětí bývá určeno konkrétním typem konvertoru. Výhodou konvertoru je, že se jedná o jednu součástku, není tedy potřeba počítat hodnoty použitých součástek a případně tyto součástky

kombinovat pro dosažení vypočítaných hodnot. Tuto možnost jsem pro práci vyhodnotil jako vhodnou a použil jsem konvertor firmy *TracoPower*. Konkrétně se jedná o typ TSR 1-2450. Tento konvertor má vstupní napětí v rozsahu 6.5V - 36V, což je pro mé zařízení plně dostačující, jelikož napájecí, pro konvertor vstupní, napětí je 12V.



Obr. 3. 1 DC/DC konvertor

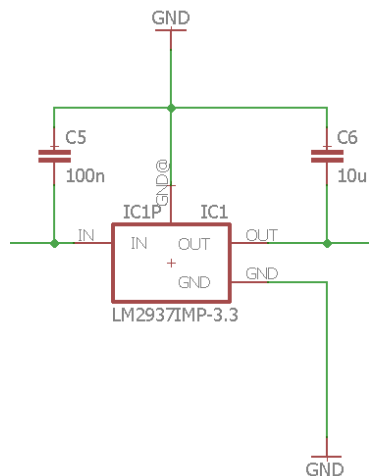
Maximální proudové zatížení výstupu je 1A, což pro tento simulátor stačí. Napětím 5V budou na desce napájeny hlavní komponenty, jako jsou inkrementální aktuátory, izolátory pro galvanické oddělení výstupu a podsvícený LCD displej.

3.1.2 Stabilizace napětí na 3,3V

Stejně jako v předchozí kapitole se jedná pouze o úpravu hodnoty stejnosměrného napětí. I v tomto případě by se dal použít napěťový dělič, ale ze stejných důvodů jsem tuto metodu nepoužil. Použití stejného konvertoru jako v předchozím případě, jenom s tím rozdílem, že výstupní napětí bude 3,3V místo 5V bylo možné. Jelikož však tímto napětím bude napájen procesor, přichází zde v úvahu použití napěťového regulátoru, který je určen přímo pro napájení mikroprocesorů.

Tyto regulátory vyrábí firma Texas Instruments. Konkrétně se jedná o regulátor typu LM2937 -3.3. Vstupní napětí je v rozsahu 4,75V až 26V a výstupní napětí je rovno 3,3V. Při výběru regulátoru je nutné spočítat si maximální odebíraný proud, aby nedošlo k překročení maximálního výstupního proudu regulátoru. Překročení tohoto proudu by mělo za následek trvalé poškození samotného regulátoru, a dále by to ovlivnilo celkovou funkci simulátoru.

Dle katalogového listu použitého displeje, je maximální odebíraný proud při napájecím napětí 3,3V 440 μ A. Odebíraný proud procesoru je pak dle katalogového listu 41 μ A. Maximální proudové zatížení výstupu regulátoru je 500mA, což je naprosto dostačující.



Obr. 3. 2 Schéma zapojení regulátoru LM2937

3.2 Processor

Zařízení pracuje samostatně, bez nutnosti připojení k počítači, proto je zapotřebí komponent, který bude celé zařízení řídit. Řídícím komponentem je mikroprocesor. Mikroprocesorů, které jsou schopny řídit toto zařízení je celá řada. Při volbě mikroprocesoru je důležité znát požadavky na komunikaci mezi mikroprocesorem a periferiemi. V tomto případě bude mikroprocesor komunikovat s periferiemi jako je LCD displej a DA převodníky přes SPI linku. Pro komunikaci s DA převodníky je využita jedna společná SPI linka. Další, samostatná, SPI linka je určena pro komunikaci s LCD displejem.

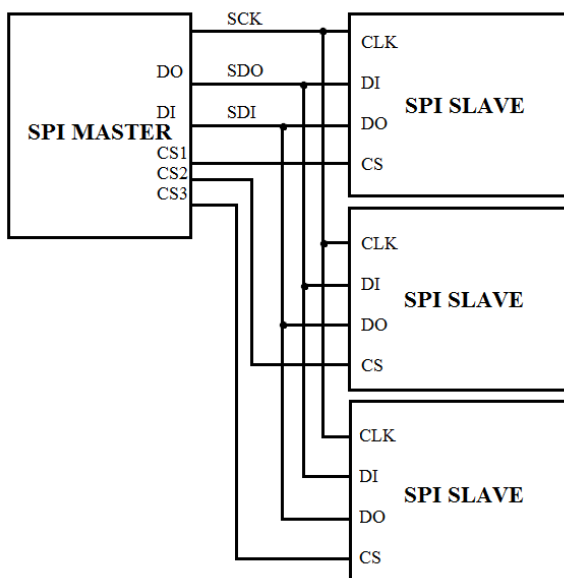
Následujícím požadavkem na mikroprocesor je dostatek vstupně výstupních portů pro připojení inkrementálních aktuátorů, které umožňují nastavování termoelektrického napětí a teploty. Mezi další periferie, které jsou připojeny na tyto porty, patří například mikrosplínače pro změnu inkrementu. Mikrosplínače zaberou celkem 8 vstupně výstupních portů, jelikož pro každý kanál jsou použity dva mikrosplínače.

Dle těchto požadavků jsem zvolil mikroprocesor firmy MicroChip. Konkrétně se jedná o mikroprocesor PIC32 - MX695F512L-80I/PT. Jedná se o 32bitový mikroprocesor. Tento mikroprocesor disponuje programovou pamětí o velikosti 512KB, a navíc ještě 12KB *bootovací flash* paměti. Datová paměť mikroprocesoru je 128KB. Dále mikroprocesor obsahuje čtyři komunikační linky SPI, které jsou pro toto zařízení velice důležité. Zvolený mikroprocesor má pouzdro s celkem 100 piny.

3.2.1 Serial Peripheral Interface (SPI)

Sběrnice SPI je jedna ze sériových externích sběrnic sloužících pro komunikaci dvou či více prvků, přičemž jeden z nich je tzv. řadič sběrnice (master) a ostatní prvky jsou podřazené, označované jako *slave*. Master obsahuje generátor hodinového signálu, který je propojen se všemi ostatními prvky, aby byl zajištěn synchronní přenos dat. Vodič, který přenáší hodinový signál,

je označován zkratkou *SCK*. Kromě vodiče s hodinovým signálem, je zde ještě dvojice vodičů, které jsou označovány zkratkami *MISO* (*Master In, Slave Out*) a *MOSI* (*Master Out, Slave In*), pomocí nichž se obousměrně přenáší data. Posledním signálem, který sběrnice SPI využívá, je signál *CS* (*Chip Select*), jenž slouží k výběru daného prvku v režimu *slave*.



Obr. 3. 3 *SPI master a slave*

Sběrnice SPI se používá v mnoha různých systémech, a to právě kvůli své jednoduchosti. Především se tato sběrnice využívá pro komunikaci s textovými i grafickými LCD panely, A/D a D/A převodníky, hodinami reálného času (*RTC*) atd. Podpora pro SPI je zabudována přímo do mnoha mikroprocesorů, kterým je mimo jiné i procesor použitý v této práci, tedy PIC32. Piny sběrnice jsou kompatibilní s technologií *TTL*. Sběrnice SPI je zjednodušeně řečeno několik externě propojených posuvných registrů, přičemž posun je řízen jednotným hodinovým signálem. Další výhodou je obousměrný přenos dat, čímž odpadá nutnost přepínání mezi vysíláním a příjmem. Hodinový takt, z něhož se odvozuje bitová rychlost (*bitrate*), se pohybuje v rozsahu 1 MHz až 10 MHz. Při dodržení dostatečně malé kapacity spoje je možné použít i frekvenci až 70 MHz.

Mezi nevýhody sběrnice SPI patří hlavně to, že pouze jeden prvek může být master. Pomocí SPI je možné přenášet data pouze na kratší vzdálenosti, toto je způsobeno jak nutností synchronizace hodinového signálu s přenášenými daty, tak i absencí signálu *ACK* (*acknowledge*), který slouží k potvrzení příjmu dat a řídí tak rychlost přenosu dat v případě, že některé zařízení nedokáže data rychle zpracovat.

Princip komunikace je velice jednoduchý. Master generuje hodinové impulsy. Pomocí hodinových impulsů se provede synchronizace pro vysílání i příjem dat. U většiny zařízení je možné nakonfigurovat polaritu hodin i to, zda se má posun provést při sestupné či vstoupné hraně hodinového signálu – důvod je uveden v předchozí kapitole. Čtyři možné tvary hodinového signálu jsou zobrazeny na osmém obrázku. Polarita hodinového signálu se určuje konfiguračním

bitem *CKP*. Další konfigurační bit *CKE* určuje, zda se platná data oznamují náběžnou či sestupnou hranou hodinového signálu. [12]

3.3 LCD display

Na DPS je umístěn také LCD displej. Při výběru displeje bylo nutné si uvědomit, co všechno se bude na displeji zobrazovat. V mém případě se na displeji bude zobrazovat termoelektrické napětí nebo simulovaná teplota termočlánku, což je podmíněno volbou uživatele. Simulovaná teplota nebo termoelektrické napětí je zobrazováno pro každý kanál zvlášť. Jelikož simulátor umožňuje simulovat 4 termočlánky najednou, je potřeba aby LCD display měl 4 řádky, především z důvodu přehlednosti. Na každém řádku je zobrazováno označení kanálu a příslušné napětí nebo teplota. Teplota je zobrazována ve stupních Celsia na jedno platné desetinné místo. Termoelektrické napětí je zobrazováno v milivoltech na 1 platné desetinné místo (Obr. 3. 4). Simulovaná teplota termočlánku je zobrazována rovněž na 1 platné desetinné místo (Obr. 3. 5). Jak můžeme vidět na příkladu, řádek má celkem 10 znaků. Celková velikost displeje je tedy 4x10 znaků.

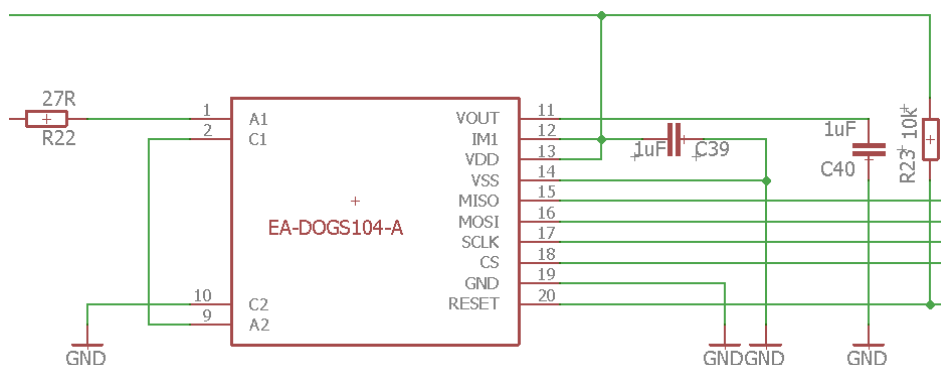
T	C	1	±	3	1	.	1	m	V
---	---	---	---	---	---	---	---	---	---

Obr. 3. 4 Zobrazení napětí na displeji

T	1	±	0	4	5	0	.	3	C
---	---	---	---	---	---	---	---	---	---

Obr. 3. 5 Zobrazení teploty na displeji

Dále je při výběru displeje nutné určit způsob komunikace a podle toho vybrat daný displej. Jelikož s displejem bude komunikovat procesor, je potřeba vybrat sběrnici, kterou obsahuje právě zvolený procesor. Jak bylo popsáno výše, mikroprocesor komunikuje s připojenými periferiemi přes sběrnici SPI. Z tohoto důvodu byl vybrán displej EA DOGS104-A, který umožňuje komunikaci pře SPI.



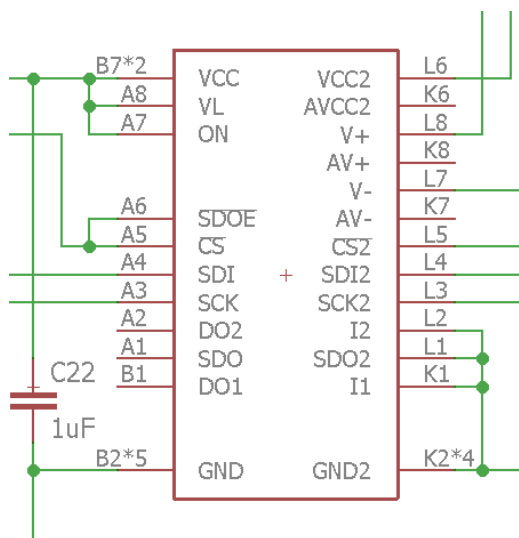
Obr. 3. 6 Schéma zapojení LCD displeje

Jelikož knihovny jednotlivých součástek v prostředí *Eagle* neobsahují schéma vybraného displeje, bylo nutné vytvořit nový knihovní prvek a schéma displeje nakreslit. Dle údajů z katalogového listu displeje byl pak tento knihovní prvek vytvořen.

LCD displej má celkem 14 pinů. Piny A1,A2,C1 a C2 slouží pro podsvícení displeje a jsou připojeny na napájecí větev 5V. Zbývajících 10 pinů slouží pro napájení displeje a komunikaci pomocí SPI linky.

3.4 Izolátory

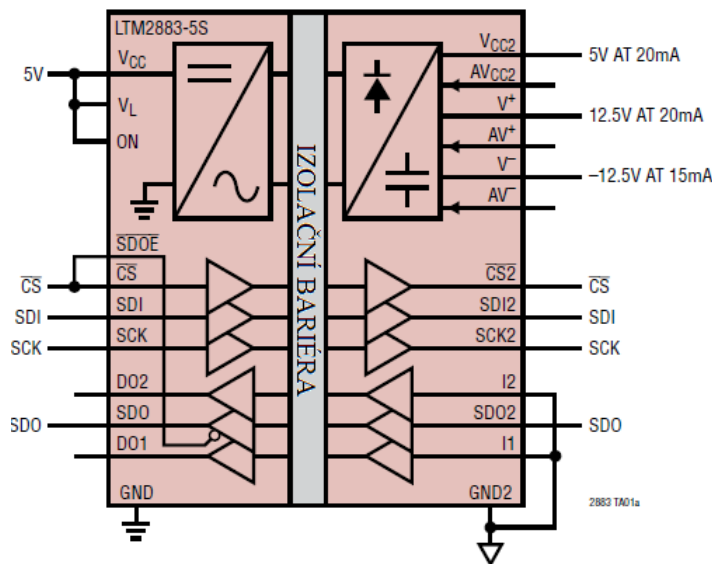
Výstupní část celého obvodu slouží pro konečnou úpravu digitálního signálu z procesoru. V první řadě je zapotřebí výstupní část galvanicky oddělit od té vstupní. Pro galvanické oddělení jsou použity izolátory.



Obr. 3. 7 Schéma zapojení izolátoru

Jelikož mikroprocesor využívá komunikaci přes sběrnici SPI, je nutné zvolit izolátory, které jsou kompatibilní s SPI. Dalším parametrem pro výběr izolátoru bylo napájecí napětí. Jak již bylo

zmíněno výše, na desce jsou rozvedeny dvě napěťové úrovně, a sice 5V a 3,3V. V rámci těchto požadavků byl vybrán izolátor typu LTM2883. Tento izolátor je šesti kanálový mikromodul s pouzdrem BGA.



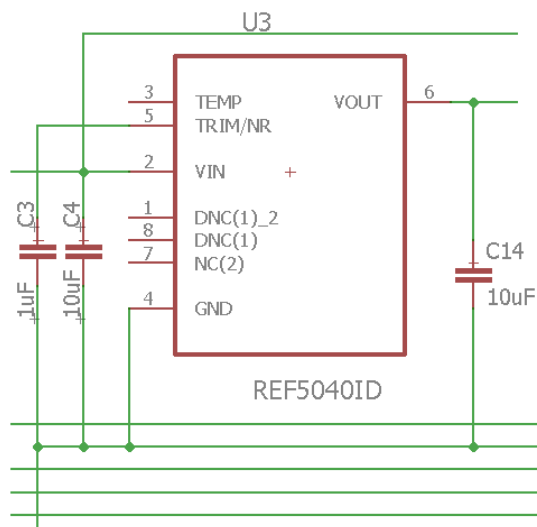
Obr. 3. 8 Vnitřní struktura izolátoru

Izolátor umožňuje využít výstupní napětí hned v několika napěťových úrovních. Výstupní pin V_{CC2} disponuje napětím 5V, což je napájecí napětí izolátoru, ale toto napětí je galvanicky oddělené. Další výstupní napětí izolátoru je $\pm 12,5V$. Toto napětí je použito pro napájení operačního zesilovače. Piny tohoto napětí jsou označeny V_+ a V_- . Izolátor obsahuje 6 kanálů a není potřeba dalších komponent. Pouzdro izolátoru je BGA.

3.5 Napěťová reference

Napěťová reference je dalším prvkem výstupní části obvodu. Reference zde slouží k udržení stálého a neměnného napětí. Napětí reference je přivedeno na referenční vstup DA převodníku a tím určuje maximální rozsah výstupního napětí DA převodníku. Rozsah vstupního referenčního napětí pro DA převodník je od 2V do VDD, což je napájecí napětí převodníku. Napájecí napětí DA převodníku je 5V. Z tohoto důvodu byl vybrán typ reference REF5040ID, jehož výstupní napětí je rovno 4,096V. Výrobce této reference je firma Texas Instruments.

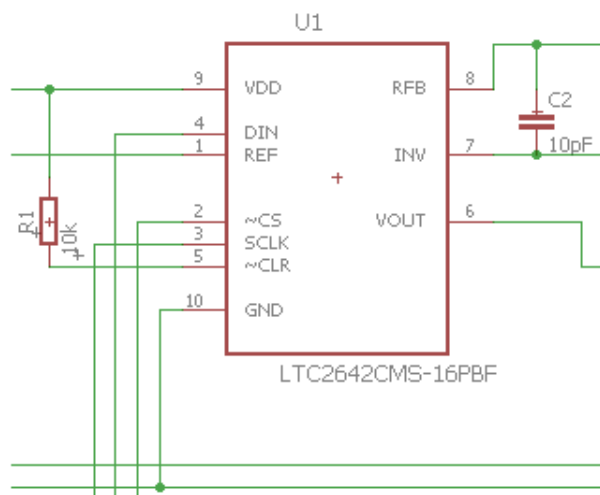
Mezi hlavní přednosti této reference patří především nízký šum a malý drift. Další silnou stránkou je také velká proudová zatížitelnost výstupu. Maximální výstupní proud může být až 10mA. Přesnost této reference je dle katalogového listu 0,1%. Mezi poslední výhody napěťové reference REF5040ID patří rozsah pracovní teploty. Rozsah pracovní teploty je udáván od $-40^{\circ}C$ do $+125^{\circ}C$.



Obr. 3. 9 Schéma zapojení reference

3.6 DA převodník

Nejdůležitějším prvkem výstupní části simulátoru je DA převodník. Jedná se o převodník, který převádí digitální signál na analogový. DA převodníky převádí signály s určitým krokem. Čím menší je krok, tím je převodník přesnější. Velikost kroku je závislá na počtu bitů převodníku a na vstupním referenčním napětí. Jak již bylo zmíněno výše, referenční napětí, které nám udává napěťová reference je rovno 4,096V.



Obr. 3. 10 Schéma zapojení DA převodníku

Počet kroků lze vypočítat následujícím vztahem:

$$x = 2^n - 1 = 2^{16} - 1 = 65535 \text{ kroků} \quad (3.1.)$$

kde: n je počet bitů

Z příkladu vyplývá, že při použití 16 bitového převodníku, budeme mít k dispozici 65535 kroků. Jakmile známe počet kroků, můžeme vypočítat přesnost kroku dle následujícího vztahu:

$$x = \frac{\pm V_{REF}}{\text{počet kroků}} = \pm \frac{4,096}{65535} = \pm 0,0000625V = \pm 0,0625mV/\text{krok} \quad (3.2.)$$

Po dosazení známých hodnot do výše uváděného vztahu jsme vypočítali, že jeden krok DA převodníku je roven 0,0625mV. Tato změna na krok, je pro tuto aplikaci dostačující, jelikož na displeji zobrazujeme termoelektrické napětí pouze na jedno desetinné místo.

Při výběru DA převodníku bylo opět důležité napájecí napětí převodníku, kompatibilita s SPI, a v neposlední řadě co největší počet kroků. Nakonec byl vybrán převodník LTC2642, který je kompatibilní s SPI. Napájecí napětí je v rozsahu 2,7 až 5,5V a převodník je 16 bitový, tedy umožňuje 65535 kroků, jak již bylo vypočítáno v příkladu výše (3.1.). Maximální frekvence, se kterou je DA převodník schopen pracovat je 50MHz, což je dostačující, jelikož na desce je osazen oscilátor o frekvenci 8MHz.

Převodník v této aplikaci plní nejdůležitější funkci. Z procesoru po SPI lince přichází na vstup převodníku binární data. Tato data jsou následně převedena na analogovou hodnotu. Velikosti analogové hodnoty, tedy výstupního napětí převodníku, odpovídá vztah:

$$V_{OUT} = \left(\frac{k}{2^{n-1}} - 1 \right) V_{REF} \quad (3.3.)$$

kde: k je decimální ekvivalent vstupního binárního kódu

n je počet bitů DA převodníku

V_{REF} je referenční napětí

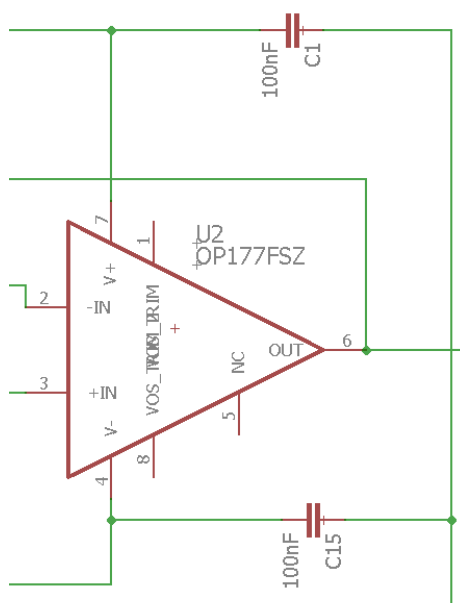
Toto napětí je v rozmezí $0V - V_{REF}$, tedy $0V - 4,096V$. Termoelektrické napětí generované reálným termočlánkem je však v řádech milivoltů a je tedy potřeba jej dále upravit, tedy posunout o řád níže.

3.7 Operační zesilovač

Jak již bylo zmíněno výše, výstupní napětí z DA převodníku je pouze v kladných hodnotách. Reálný termočlánek však generuje, při záporných hodnotách teploty, záporné termoelektrické napětí. Aby simulátor umožňoval generovat i záporné termoelektrické napětí, je potřeba na výstup DA převodníku zapojit právě zmíněný operační zesilovač.

Na invertující vstup je přivedeno invertované referenční napětí, které je invertováno přímo v DA převodníku. Na neinvertující vstup je přivedeno kladné výstupní napětí DA převodníku. Do zpětné vazby operačního zesilovače je pak přivedeno kladné referenční napětí. Výstupní napětí operačního zesilovače je poté v rozsahu $\pm V_{REF}$.

Pro tento simulátor byl vybrán operační zesilovač typu OP177ISZ. Onen operační zesilovač má hned několik předností. Jelikož je generované termoelektrické napětí řádově v milivoltech, je potřeba, aby byl OZ co nejpřesnější. Hlavní složka, která ovlivňuje výstupní napětí OZ je napěťový offset. Napěťový offset operačního zesilovače OP177ISZ je maximálně $25\mu\text{V}$, při provozní teplotě 25°C . Veličina, která s napěťovým offsetem OZ úzce souvisí je tzv. napěťový drift. Napěťový drift udává změnu napěťového offsetu v závislosti na změně teploty o jeden stupeň Celsia. Pro tento typ OZ je výrobcem udáván napěťový drift maximálně $0,1\mu\text{V}/^\circ\text{C}$. Pracovní teplota OZ je v rozmezí od -40°C do $+85^\circ\text{C}$. Operační zesilovač bude napájen $\pm 12,5\text{V}$. Napájecí napětí je přivedeno z izolátoru, konkrétně z pinů V_+ a V_- .



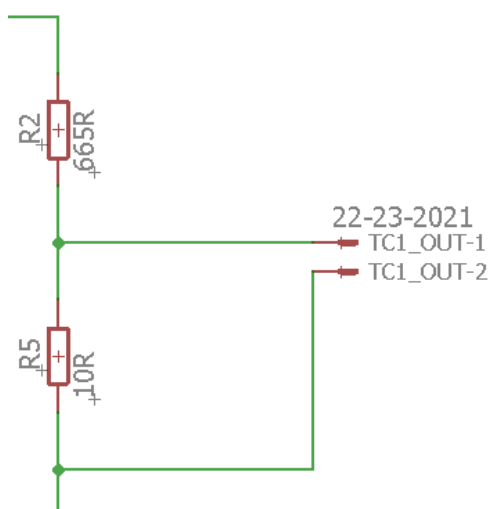
Obr. 3. 11 Schéma zapojení operačního zesilovače

3.8 Napěťový dělič

Poslední prvek výstupní části simulátoru je napěťový dělič. Jak již bylo zmíněno výše, výstupní napětí zesilovače je řádově v jednotkách voltů, ale termoelektrické napětí generované reálným termočlánkem je v řádově v milivoltech. K úpravě napětí na správný řád zde slouží právě napěťový dělič.

Přesnost napěťového děliče je přímo ovlivněna přesností použitých rezistorů. V tomto případě jsou použity rezistory s přesností 0,1%. Hodnoty rezistorů jsou pro R_2 665 Ω a pro R_5 10 Ω . Dělicí poměr děliče vypočítáme dle následujícího vztahu:

$$a = \frac{R_5}{R_2 + R_5} = \frac{10}{665 + 10} = 0,0148 \quad (3.4.)$$

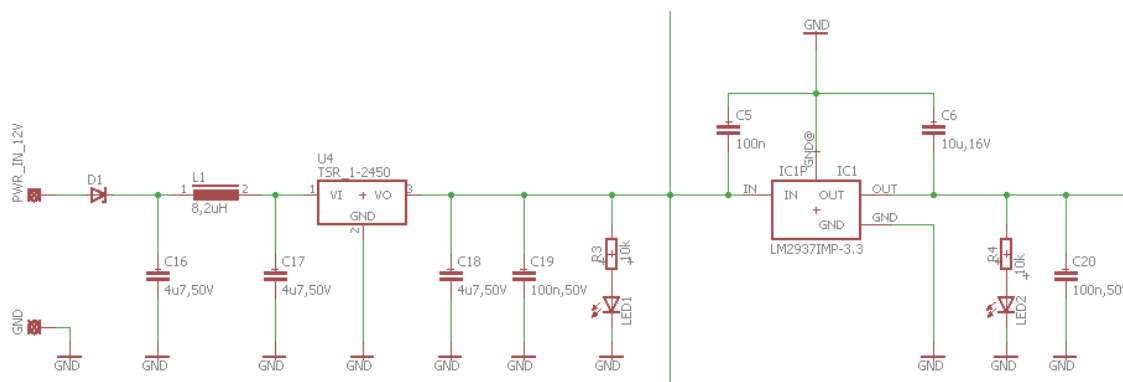


Obr. 3. 12 Schéma zapojení napěťového děliče

3.9 Schéma zapojení

Jakmile jsem si vybral prostředí, ve kterém budu navrhovat DPS, prvním krokem bylo navrhnout obvodové schéma celého simulátoru. Od zadavatele práce, firmy Continental, mi bylo poskytnuto vzorové schéma, ze kterého jsem vycházel.

V první řadě byla nakreslena vstupní část obvodu, tedy část napájecí. Jak již bylo zmíněno v teoretickém rozboru, napájecí napětí celé DPS je stejnosměrných 12V. Jelikož by mohlo dojít k náhodnému přepólování vstupního napájecího napětí, je zde implementována dioda, která právě proti přepólování napájecího napětí obvodu simulátoru chrání.



Obr. 3. 13 Napájecí část obvodu

Hned dalším krokem byla řídicí část obvodu, což není nic jiného než mikroprocesor. Dle katalogového listu mikroprocesoru, byly jednotlivé komponenty připojovány k procesoru. Při kreslení řídicí části obvodu, bylo velmi důležité připojit komponenty na správné piny procesoru. Pasivní součástky jsou připojeny k procesoru dle katalogového listu, ale při kreslení SPI linek pro řízení displeje a DA převodníků bylo nutné určit, která linka bude komunikovat s displejem a která s DA převodníky. Jestliže by došlo k chybnému připojení některého z pinů, celá DPS by nefungovala správně a mohlo by ve výsledku dojít k nevratnému poškození jak procesoru, tak i dalších komponentů.

Procesor umožňuje komunikaci přes 4 nezávisle na sobě SPI linky. Pro řízení výstupní části obvodu, tedy DA převodníků, byla použita jedna společná linka pro všechny 4 kanály. Pro řízení displeje byla použita samostatná SPI linka.

Kromě výstupních periférií, jako byl již zmíněný display a DA převodníky, bylo zapotřebí připojit k procesoru také vstupní periférie. Nejdůležitější vstupní periférie jsou enkodéry, které umožňují uživateli nastavovat požadované termoelektrické napětí nebo teplotu. Enkodéry jsou připojeny na 5V, proto bylo důležité vybrat správné piny procesoru, které nejsou omezeny pouze na napětí 3,3V. Pro připojení enkodérů byly vybrány piny s označením RE0 – RE7.

Další vstupní periférií byl mini USB port. Tento port slouží pro komunikaci simulátoru s počítačem a umožňuje tak ovládat simulátor - například nastavovat termoelektrické napětí nebo teplotu přes aplikaci pro sériovou komunikaci, např. RealTerm. Pro připojení USB portu k procesoru jsou výrobcem přesně specifikované piny.

Jako poslední bylo nutné k procesoru připojit programovací konektor. Tento konektor má 5 pinů a stejně jako u USB portu jsou i zde určeny piny přímo výrobcem.

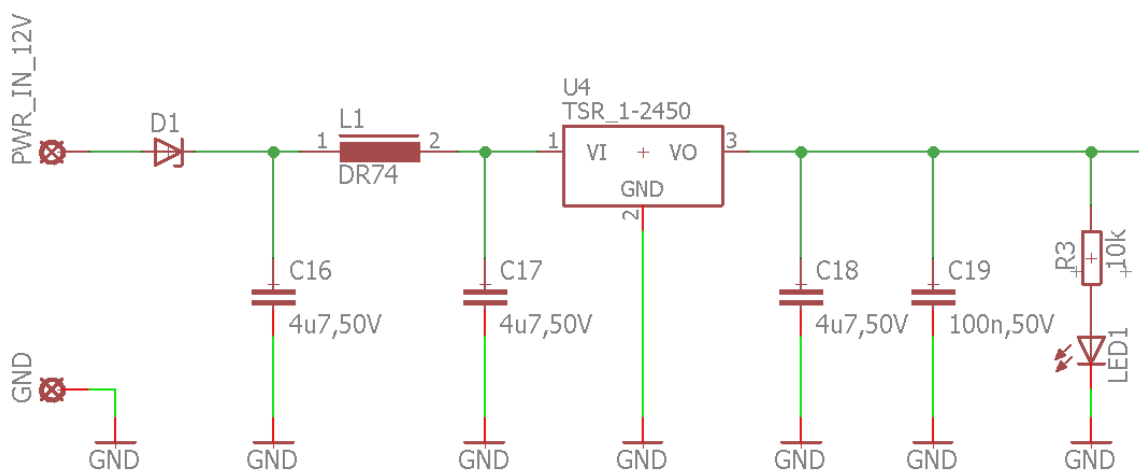
Poslední část schématu je část výstupní. Tato část má největší prioritu, jelikož zaručuje přesnost celého simulátoru. Jak již bylo popsáno v teoretickém rozboru, první prvek výstupní části je izolátor pro galvanické oddělení výstupní části obvodu. Izolátory jsou přímo připojeny

k procesoru přes společnou SPI linku. K izolátoru je následně připojena napěťová reference, dále DA převodník a operační zesilovač.

Při kreslení výstupní části obvodu bylo důležité správně nakreslit a připojit signálovou zem GND. Všechny komponenty výstupní části musí být připojeny na zem izolátoru. V žádném případě nemohou být komponenty připojeny na stejnou zem jako ostatní komponenty na DPS. V případě, že by se tak stalo, funkce izolátoru by neměla žádný smysl a výstupní část obvodu by nebyla galvanicky oddělená. Stejně tak bylo důležité všechny komponenty, kromě těch ve výstupní části, připojit na společnou zem.

Velice důležité bylo při kreslení celého schématu kontrolovat správnost připojení každého pinu, každé součástky. Kdyby došlo k nekorektnímu připojení pinu, mělo by to velký vliv na funkčnost celého simulátoru. Dále by při návrhu DPS a rozmisťování součástek na DPS nedošlo k vytvoření vodivé cesty.

Pro kontrolu správného připojení jednotlivých signálových větví, ať už se jedná o signálovou zem nebo o jednotlivé napěťové úrovně, slouží v prostředí *Eagle* nástroj *Show*. Tento nástroj umožňuje označit danou signálovou větev, která se v celém schématu zvýrazní včetně pinů na které je tento signál připojen (Obr. 3. 14). Aby však tento nástroj fungoval správně, je zapotřebí dodržovat stejné pojmenování jednotlivých signálů, například pro signálovou zem označení GND.



Obr. 3. 14 Použití funkce *SHOW*

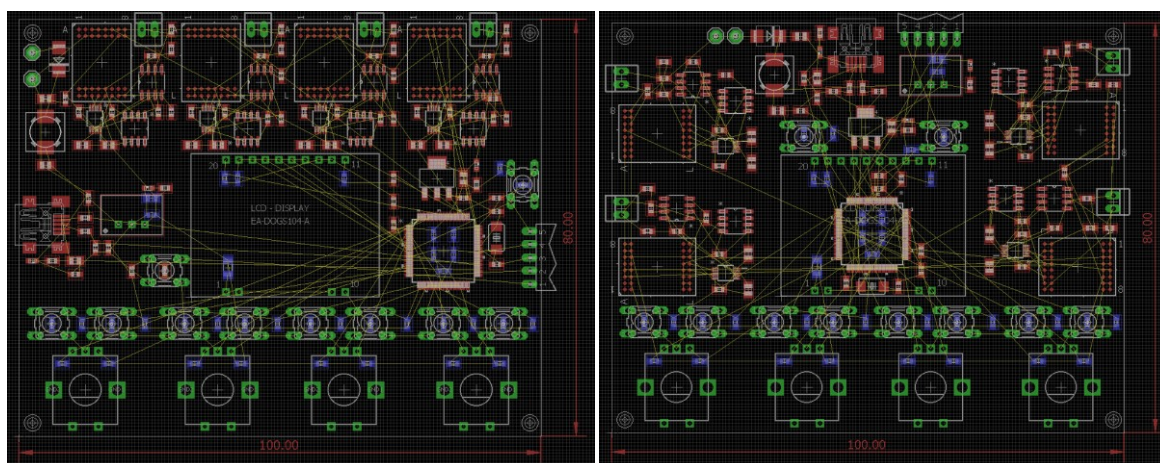
V poslední řadě bylo nutné dodržovat základní pravidla pro tvorbu schémat. Popisky jednotlivých komponentů umisťovat tak, aby se nepřekrývaly a byly čitelné. Veškeré signálové vodiče bylo nutné kreslit tak, aby se co nejméně křížily. U křížení vodičů bylo důležité vyznačovat vodivé uzly.

3.10 Rozmístění součástek na DPS

Nakreslit obvodové schéma je základní stavební kámen při tvorbě vlastní DPS. Další krokem je rozmístění součástek na samotnou desku. V první řadě je třeba určit rozměry desky a počet vrstev. V tomto případě je zvolena deska o velikosti 10 x 8 cm. Navrhovaná deska je oboustranná, a má dvě vrstvy.

Jakmile jsou určeny rozměry a počet vrstev, přichází na řadu rozvržení celé desky. Nejprve je vhodné na desku umístit hlavní součástky, jako jsou v tomto případě procesor, displej a další. Pasivní součástky jako jsou jednotlivé rezistory a kondenzátory je lepší si nechat na později, až budou přibližně rozmístěny hlavní součástky.

V rámci návrhu této desky byly vyzkoušeny dva hrubé návrhy. V prvním návrhu byl displej umístěn do středu desky. Procesor byl po pravé straně displeje. Prostor pro enkodéry a mikropínače byl stejný v obou případech, a sice ve spodní části desky. Výstupní část pro jednotlivé kanály, která obsahuje izolátor, DA převodník, operační zesilovač atd. byla umístěna v horní části desky. Rozvržení desky můžeme vidět na obrázku níže (Obr. 3. 15).



Obr. 3. 15 Návrh DPS - varianta A; varianta B

První změna ve variantě B je umístění procesoru. Z pravé strany DPS byl procesor přemístěn pod LCD displej. Přesunutím procesoru pod displej se uvolnilo místo na pravé straně DPS pro izolátory a ostatní součástky výstupní části. Stabilizátor napětí a USB port byl přemístěn do horní části DPS místo izolátorů. Výstupy jednotlivých kanálů byly rozděleny na dvě dvojice. Jedna dvojice na pravou stranu a druhá dvojice na stranu levou. Varianta B umožňuje větší přehlednost a více místa na DPS. Více místa znamená, že součástky nemusí být tak blízko u sebe a tím vznikne i větší prostor pro vodivé cesty. Z těchto důvodů byla zvolena varianta B.

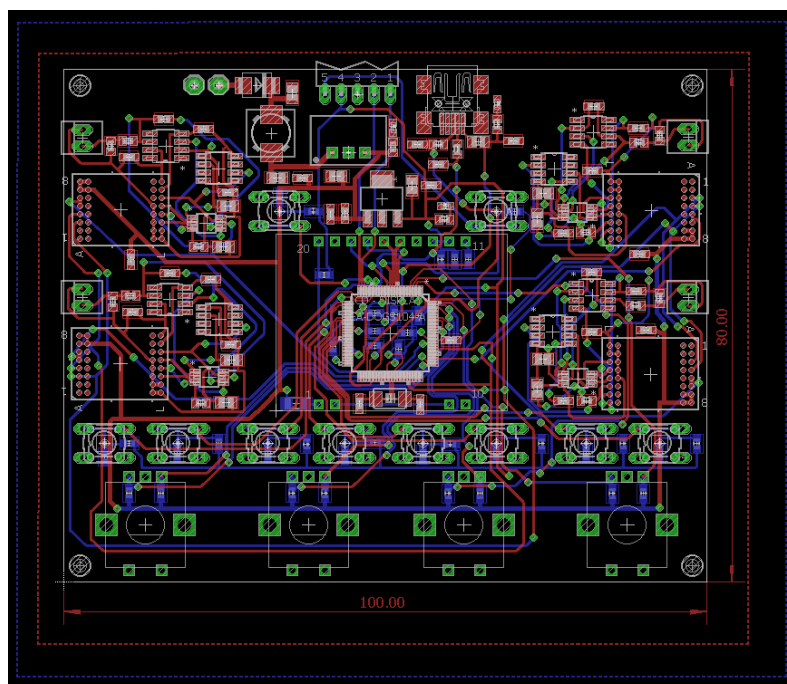
Jakmile jsou tzv. nhrubo rozmístěny součástky, můžeme přejít k vytváření vodivých cest. Hlavní součástky, se kterými již nechceme pohybovat, můžeme pomocí funkce *lock* zamknout. Při tvorbě vodivých cest je možné použít funkci *autorouter*. Tato funkce je však při návrhu větších desek nevhodná, jelikož nedokáže žádnou součástku dle potřeby přesunout nebo otočit. Z tohoto

důvodu není funkce autorouter schopna, u složitějších DPS vytvořit všechny vodivé cesty. V tomto případě bylo při použití funkce autorouter vytvořeno pouze 67% cest. Další nevýhodou funkce je, že vytvořené vodivé cesty, je potřeba dále vizuálně upravit.

Další možností, jak vytvořit vodivé cesty, je použití manuální funkce *route*. Pomocí této funkce si uživatel vytváří každou cestu zvlášť. Při manuální tvorbě vodivých cest je vhodné si nejprve navrhnut napájecí větve a zem. Pro uzemnění všech součástek je použit polygon. Tato funkce nám umožní tzv. vylítí země, což způsobí propojení všech zemních pinů a svorek na desce. Jelikož je deska oboustranná, bylo zapotřebí vytvořit celkem dva polygony. Při tvorbě napájecích větví je potřeba zvolit větší tloušťku cest. V tomto případě byla pro napájecí větve zvolena tloušťka cesty 0,8128mm a 0,6096mm.

Jakmile jsou vytvořeny napájecí větve, přichází na řadu vytvoření cest mezi jednotlivými součástkami. Pro jednotlivé cesty je vhodné používat menší tloušťky než u cest napájecích větví. V tomto projektu byly použity cesty o velikosti 0,4064mm; 0,3048mm a 0,254mm. Nejmenší tloušťka cesty byla použita například pro připojení jednotlivých pinů procesoru.

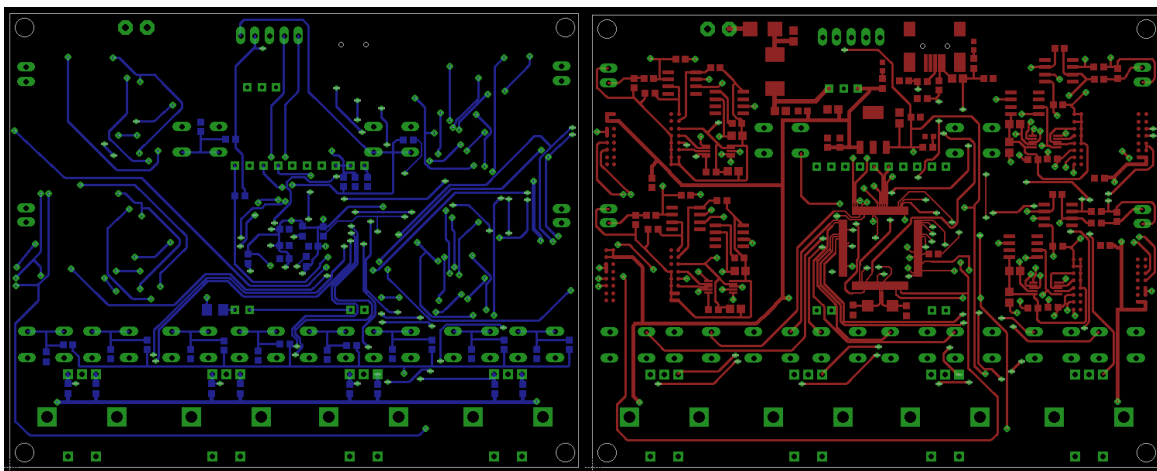
Při tvorbě vodivých cest je důležité dodržovat základní pravidla. Tím nejdůležitějším pravidlem je, že se žádná cesta nesmí křížit s jinou. Dále pak musí být každá cesta vedena kolem součástky s dostatečnou mezerou. To samé platí i o mezeře mezi jednotlivými cestami. Tyto základní pravidla hlídá funkce DRC (*Design Route Check*).



Obr. 3. 16 Konečná podoba DPS

Nakonec se musí finální návrh desky (Obr. 3. 16) upravit na masku pro osvětlení fotocitlivé desky plošných spojů. Úprava schématu se provede pouze ve správci hladin desky. Správce hladin

umožňuje zobrazovat a skrývat jednotlivé hladiny. Zobrazením hladin pro vodivé cesty a piny součástek dostaneme osvitovou masku (Obr. 3. 17). Zbylé hladiny zůstanou skryty.

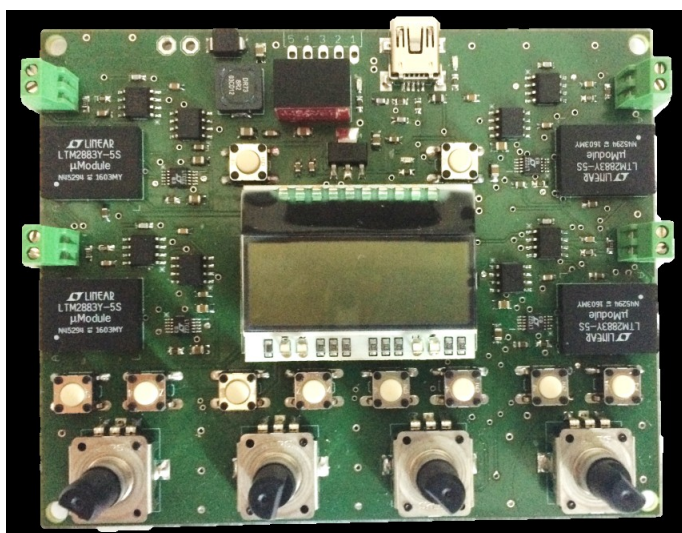


Obr. 3. 17 Osvitová maska

3.11 Osazení DPS

Jakmile je hotový návrh desky, je potřeba desku vyrobit. Deska byla vyrobena ve firmě Gatema. Po vyrobení bylo nutné desku osadit jednotlivými součástkami. Jako první se osazovaly součástky, jejichž pájení je složitější, jako například procesor a izolátory. Jako další se osazovaly součástky výstupní části simulátoru, jako jsou DA převodníky, operační zesilovač a další.

Po osazení všech integrovaných obvodů, se na desku osadily vývodové součástky, a sice inkrementální aktuátory, mikrospínače apod. Jako poslední byly na desku osazeny pasivní součástky. Použitá pouzdra pasivních součástek byla převážně 0603 a 0805.



Obr. 3. 18 Plně osazená DPS

Osazování desky probíhalo u zadavatele práce, firmy Continental. Jelikož jsou SMD součástky velmi malé a jejich velikosti je řádově v desetinách až jednotkách milimetru, je při osazování těchto součástek nezbytná velká preciznost.

U osazování součástek je velmi důležité kontrolovat správné zapájení. V případě špatného zapájení vznikne tzv. studený spoj. Studený spoj znamená, že součástka není vodivě spojena s pájecí ploškou na DPS a není tak zajištěna správná funkčnost součástky a tedy i celé vodivé cesty, na které je daná součástka umístěna. Na obrázku níže je vyobrazena již kompletně osazená deska.

4 Implementace softwaru

V této kapitole je popsána architektura zdrojového kódu, jeho celkový vývoj a následná implementace. Dále tato kapitola popisuje bitovou konfiguraci mikroprocesoru a v neposlední řadě jednotlivé funkce, které obsahuje zdrojový kód. Jakmile je deska plně osazena, je zapotřebí ji tzv. oživit. To znamená sestavit jednoduché obslužné programy pro jednotlivé periferie. Při ožívování desky se zjistí, zda všechny komponenty umístěné na desce fungují správně. Také se zde dále otestuje komunikace mezi jednotlivými komponenty.

4.1 Softwarová architektura

Pojem softwarová architektura je v dnešní době definován velkým množstvím různých definic. Jednotlivé definice jsou sice odlišné, ale jádro těchto vymezení je stejné. Ve své podstatě softwarová architektura popisuje stavbu celého zdrojového kódu.

Během plánování architektury kódu jsem sestavil dva odlišné návrhy. První z nich byl sestavit kód v jedné *while* smyčce. Smyčka *while* běží kontinuálně, což znamená, že se neustále opakuje dokola, tedy jakmile dojde na konec kódu, vrátí se na začátek a program probíhá znovu. V této smyčce byly vloženy všechny podmínky a funkce pro jednotlivé periferie. V kontinuálně běžící smyčce jsou neustále vyhodnocovány podmínky. Jednotlivé podmínky se vyhodnocují popořadě tak, jak jsou zapsány ve zdrojovém kódu. Jakmile je daná podmínka splněna, vykoná se její obsah. Podmínky mohou obsahovat jeden nebo více příkazů nebo také odkaz na danou funkci, která se při splnění dané podmínky vykoná.

Druhou variantou bylo sestavit program na základě tzv. přerušení. Sestavený program, ve kterém se využívá přerušení, obsahuje stejně jako v předchozím případě, smyčku *while*. Tato smyčka však na rozdíl od předchozího případu neobsahuje všechny podmínky a funkce ve svém těle. Na základě vstupního signálu, například stisku tlačítka, dojde k přerušení hlavní smyčky a vykoná se obslužná funkce pro stisk tlačítka. Jakmile se tato funkce vykoná, obslužná funkce se uzavře a hlavní smyčka bude dále pokračovat, dokud zase nepřijde signál přerušení.

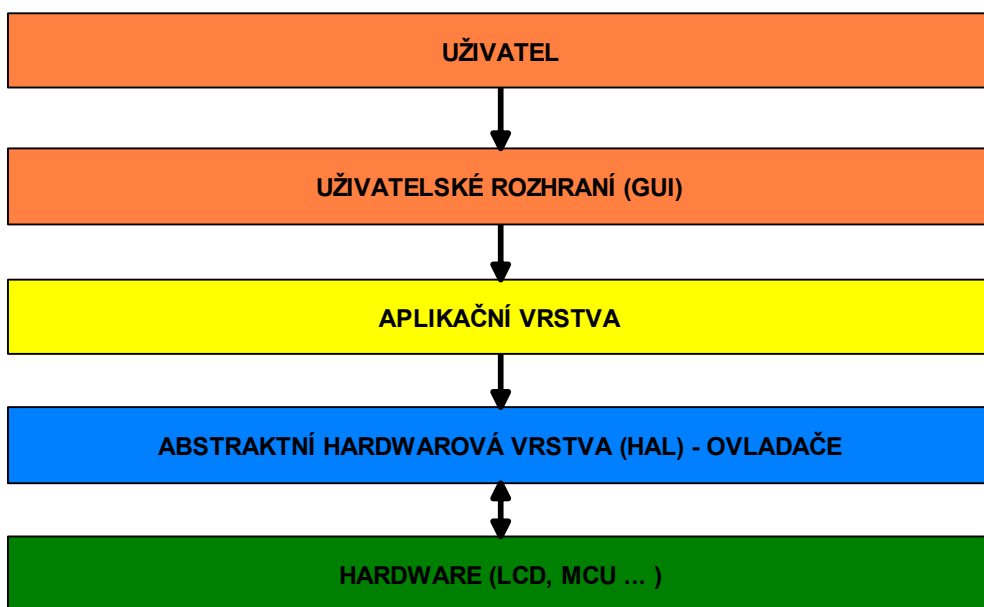
V tomto případě jsem se nakonec rozhodl postavit program na základě jedné *while* smyčky. Tato metoda má oproti metodě s přerušením jednu zásadní nevýhodu. Jelikož se podmínky vyhodnocují popořadě, tak jak jsou napsány v kódu, může se stát, že mikroprocesor nezareaguje na stisk tlačítka, jehož funkce je napsána na začátku celého kódu. Jelikož je však mnou zvolený mikroprocesor velmi rychlý, tato situace by vůbec neměla nastat.

Nedílnou součástí softwarové architektury je také blokový diagram. Blokový diagram umožňuje graficky zobrazit vazby mezi jednotlivými vrstvami. Jako první se v blokovém diagramu nachází uživatelská vrstva. Uživatelská vrstva obsahuje uživatelské rozhraní. Uživatelské rozhraní slouží ke komunikaci mezi zařízením a uživatelem.

Jako další je zde aplikační vrstva. Tato vrstva zde slouží jako brána, která zprostředkovává přístup aplikace do abstraktní hardwarové vrstvy a tím umožňuje komunikaci aplikace

s periferiemi. Abstraktní hardwarová vrstva je v pořadí další vrstva blokového diagramu. V této vrstvě jsou umístěny ovladače periférií. Mikroprocesor zpracovává veškeré informace získané přes uživatelské rozhraní od uživatele, například nastavení 20mV na kanál č. 3. Jakmile mikroprocesor tento požadavek zpracuje, musí obsloužit jednotlivé periferie, jako je například DA převodník, pomocí kterého nastaví na výstup dané napětí a LCD displej, na kterém může uživatel vidět, jaká hodnota napětí je na daném kanále nastavena. Aby však mohl mikroprocesor dané periferie obsloužit, musí s nimi umět pracovat a komunikovat. Pro komunikaci a práci mikroprocesoru s periferiemi jsou zde tzv. ovladače. Ovladač je zjednodušeně komunikační protokol. Bez ovladačů by mikroprocesor nevěděl, jak k dané periférii přistupovat a pracovat s ní.

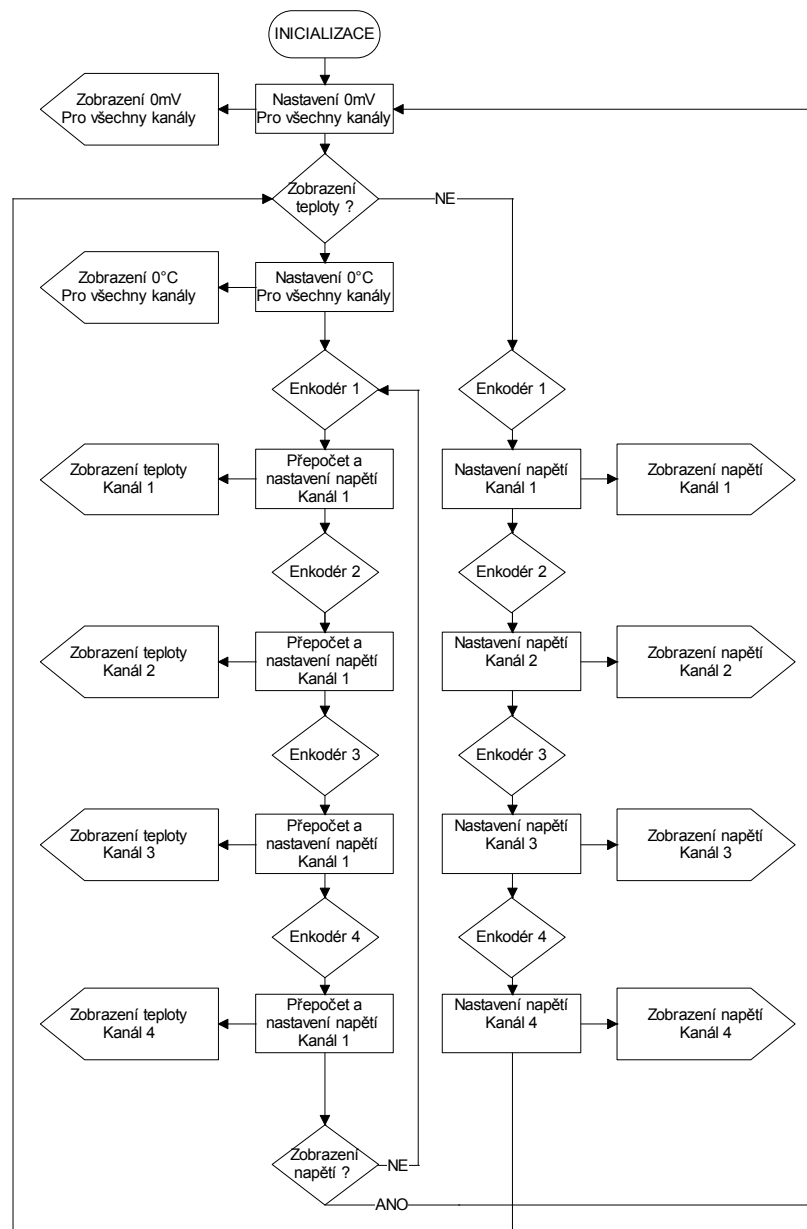
Poslední část blokového diagramu je hardwarová vrstva. Do této vrstvy spadají periferie. Periferie mohou být obousměrné, což znamená, že mohou data od mikroprocesoru jak přijímat, tak je i posílat zpět do mikroprocesoru. Periferie přistupují do abstraktní hardwarové vrstvy, aby byly schopny mezi sebou komunikovat.



Obr. 4. 1 *Blokový diagram vrstev*

4.1.1 Vývojový diagram řídicího softwaru

Vývojový diagram slouží ke grafickému znázornění jednotlivých kroků a algoritmů celého řídicího kódu. Pro zobrazení jednotlivých kroků jsou ve vývojovém diagramu použity geometrické obrazce různého tvaru. Níže zobrazený vývojový diagram (Obr. 4. 2) určuje jednotlivé kroky řídicího softwaru, který je implementován v simulátoru. Jako první krok po zapnutí simulátoru se provede inicializace simulátoru. V bloku inicializace se vykoná deklarace jednotlivých funkcí a globálních proměnných. Během inicializace se globálním proměnným přiřadí jejich počáteční hodnota, například proměnné obsahující hodnotu simulovaného napětí se vynulují.



Obr. 4. 2 Vývojový diagram

Dalším krokem po inicializaci je nastavení a zobrazení simulovaného napětí na 0mV pro všechny kanály. Po tomto kroku následuje podmínka pro zobrazení teploty, jelikož původní zobrazovaná veličina je termoelektrické napětí. Je-li podmínka splněna, stiskem tlačítka simulátor nastaví a zobrazí pro všechny kanály 0°C. Následně je simulátor v režimu "teplota" a pomocí enkodérů lze nastavovat simulovanou teplotu na jednotlivých kanálech. Nastavená simulovaná teplota je zobrazena na displeji simulátoru, a je dále přepočtena na termoelektrické napětí, které je nastaveno na příslušném kanálu. Pokud nedojde opět ke stisku tlačítka pro změnu zobrazované veličiny, simulátor pracuje stále v režimu teploty.

Jestliže dojde během doby, kdy simulátor pracuje v režimu "teplota" ke stisku tlačítka pro změnu zobrazované veličiny, simulátor přejde do režimu "napětí". Hned po přechodu do režimu "napětí" simulátor nastaví na všechny kanály napětí o velikosti 0mV. Po nastavení nulového napětí, lze pomocí enkodérů toto napětí na jednotlivých kanálech měnit. Simulátor pracuje v režimu "napětí" opět po dobu, než bude znovu stisknuto tlačítko pro změnu zobrazované veličiny.

4.2 Konfigurace mikroprocesoru

Jako první krok po založení nového projektu ve vývojovém prostředí MPLAB, je bitová konfigurace mikroprocesoru. Bitovou konfigurací mikroprocesoru je myšleno povolení, zakázání nebo nastavení jednotlivých bitů mikroprocesoru. Konfigurace mikroprocesoru probíhá přímo ve vývojovém prostředí MPLAB. V záložce Run se nachází kolonka *Set Configuration Bits*. V dolní části obrazovky se zobrazí okno, kde je možné jednotlivé bity konfigurovat. Pro každý bit je zde nastavení v podobě rozbalovací nabídky, není tak možné navolit nesprávné parametry, které by byly v rozporu s použitým mikroprocesorem. Konfigurační bity jsou rozděleny do čtyř skupin.

První skupina obsahuje vstupně výstupní porty, jako jsou *Ethernet* či *USB*. Pro *Ethernet* je zde možné konfigurovat jednotlivé piny alternativně nebo ponechat původní nastavení. Dále je zde možné nastavit ovládání *USB* portu, a sice jestli bude ovládání pomocí funkce portu, nebo *USB* modulu.

Druhá skupina se týká tzv. fázového závěsu (*PLL – Phase Locked Loop*). Fázový závěs generuje výstupní signál, jehož fáze závisí na fázi referenčního signálu. Tento řídicí systém porovná fázi vstupního signálu s fázi referenčního signálu, a upraví frekvenci oscilátoru tak, aby se tyto fáze shodovaly [13]. Frekvence vstupního signálu je stejná, jako frekvence výstupního signálu. Pomocí fázového závěsu lze však generovat i několikrát větší nebo menší frekvenci, než je původní frekvence oscilátoru. V mém případě fázový závěs generuje 2x menší frekvenci, tedy 40MHz. Snížení této frekvence je provedeno z důvodu, že není potřeba, aby mikroprocesor pracoval na tak vysoké frekvenci. Pro tuto aplikaci je frekvence 40MHz naprosto dostačující.

Další skupina konfiguračních bitů je určena pro nastavení oscilátoru a *Watchdogu*. V rámci této skupiny bitů je možné nastavit zdroj systémových hodin. Zdrojem systémových hodin může být interní oscilátor mikroprocesoru nebo signál generovaný fázovým závěsem. Dále je zde možné povolit či zakázat druhý oscilátor. Druhý oscilátor je pro tuto aplikaci vypnutý. V neposlední řadě je zde možné nastavit časovač *Watchdogu*. *Watchdog* hlídá zacyklení programu v určitém čase, který je nastavitelný v této skupině bitů. Jestliže je program zacyklený déle, než je nastavený čas, *Watchdog* celý program resetuje. V mém případě je však *Watchdog* vypnutý.

Poslední skupina je zaměřena na nastavení ochrany proti přepsání zdrojového kódu. Ochrana proti přepisu je v mém případě vypnuta, jelikož může být kód v průběhu testování upravován dle potřeby. Je-li však do zařízení nahrán finální software a zařízení je předáno zákazníkovi, pak je tato ochrana zapnutá, aby zákazník nemohl řídicí software přepsat.

Po nastavení všech konfiguračních bitů vývojové prostředí MPLAB, umožňuje vygenerovat kód, který bude tuto konfiguraci obsahovat.

```
// DEVCFG3
// USERID = No Setting
#pragma config FSRSEL = PRIORITY_7 // SRS Select (SRS Priority 7)
#pragma config FMIEN = OFF // Ethernet RMII/MII Enable (RMII Enabled)
#pragma config FETHIO = ON // Ethernet I/O Pin Select (Default Ethernet I/O)
#pragma config FUSBIDIO = ON // USB USID Selection (Controlled by the USB Module)
#pragma config FVBUSONIO = ON // USB VBUS ON Selection (Controlled by USB Module)

// DEVCFG2
#pragma config FPLLIDIV = DIV_2 // PLL Input Divider (2x Divider)
#pragma config FPLLMUL = MUL_20 // PLL Multiplier (20x Multiplier)
#pragma config UPLLIDIV = DIV_2 // USB PLL Input Divider (2x Divider)
#pragma config UPLEN = ON // USB PLL Enable (Enabled)
#pragma config FPLODIV = DIV_2 // System PLL Output Clock Divider (PLL Divide by 2)

// DEVCFG1
#pragma config FNOSC = PRIPLL // Oscillator Selection Bits (Primary Osc w/PLL (XT+,HS+,EC+PLL))
#pragma config FSOSCEN = OFF // Secondary Oscillator Enable (Disabled)
#pragma config IESO = OFF // Internal/External Switch Over (Disabled)
#pragma config POSCMOD = XT // Primary Oscillator Configuration (XT osc mode)
#pragma config OSCIOFNC = OFF // CLKO Output Signal Active on the OSCO Pin (Disabled)
#pragma config FPBDIV = DIV_1 // Peripheral Clock Divisor (Pb_Clk is Sys_Clk/1)
#pragma config FCKSM = CSDCMD // Clock Switching and Monitor Selection (Clock Switch Disable, FSCM Disabled)
#pragma config WDTPS = PS4096 // Watchdog Timer Postscaler (1:4096)
#pragma config FWDTEN = OFF // Watchdog Timer Enable (WDT Disabled (SWDTEN Bit Controls))

// DEVCFG0
#pragma config DEBUG = OFF // Background Debugger Enable (Debugger is disabled)
#pragma config ICESEL = ICS_PGx1 // ICE/ICD Comm Channel Select (ICE EMUC1/EMUD1 pins shared with PGx1/FGD1)
#pragma config FWP = OFF // Program Flash Write Protect (Disable)
#pragma config BWP = OFF // Boot Flash Write Protect bit (Protection Disabled)
#pragma config CP = OFF // Code Protect (Protection Disabled)
```

Obr. 4. 3 Bitová konfigurace mikroprocesoru

4.3 Použité knihovny

Tato kapitola popisuje použité ovladače neboli knihovny, při tvorbě zdrojového kódu. V knihovně se deklarují takové funkce, které mohou sloužit např. pro matematické operace obsluhující dané periferie apod. Mnoho knihoven již bylo vytvořeno a jsou součástí vývojového prostředí MPLAB. Pokud se však používá externí periferie, která není standardní, je potřeba tuto knihovnu vytvořit. Pro aplikaci byly vytvořeny knihovny pro DA převodník, SPI a LCD displej. Tyto knihovny byly vytvořeny ve firmě Continental.

Mezi standardní knihovny, které byly použity, patří:

- GenericTypeDefs.h
 - knihovna *GenericTypeDefs* definuje různé datové typy a jejich vzájemné přetypování
- Compiler.h
 - tato knihovna obsahuje specifikace kompilátoru pro všechny typy procesoru PIC
- stdio.h
 - stdio je knihovna, která obsahuje standardní příkazy pro vstupy a výstupy

- string.h
 - knihovna *string* deklaruje textové řetězce a jejich funkce
- math.h
 - tato knihovna obsahuje matematické funkce
- USB/usb.h, USB/usb_function_cdc.h, USB/usb_config.h, USB/usb_device.h
 - tyto knihovny obsahují deklaraci funkcí pro obsluhu USB portu
- p32mx695f5121.h
 - knihovna obsahuje specifikace přímo pro daný typ mikroprocesoru

4.3.1 Knihovna SPI

Knihovna SPI obsahuje funkce pro komunikaci mikroprocesoru s periferiemi pomocí SPI linky. V hlavičkovém souboru SPI.h jsou deklarovány dvě funkce, a sice funkce pro inicializaci SPI linky a funkce pro výběr *slave* modulu. K této knihovně se však pojí ještě zdrojový kód, který tyto funkce obsahuje. V rámci této aplikace jsou využity dvě na sobě nezávislé linky SPI. Zdrojový kód SPI.c obsahuje funkce pro první použitou linku. Tato linka je využívána pro komunikaci s DA převodníky.

První funkce slouží k inicializaci SPI linky. V této funkci je nastavena komunikační frekvence a mód linky. SPI linka pracuje v módu master - *slave*. Jak již bylo zmíněno výše, modul master je vždy jen jeden a ostatní moduly jsou *slave*. V tomto případě je mikroprocesor nastaven jako master. Na konec této funkce je umístěn reset SPI linky.

```

void InitSPI1(unsigned short frequency)
{
    SPI1BRG = frequency;
    //IECOCLR = 0x03800000;// disable all interrupts
    //IFS0CLR = 0x03800000;// clear any existing event
    //IPC5CLR = 0x1f000000;// clear the priority
    //IPC5SET = 0x0d000000;// Set IPL=3, Subpriority 1
    SPI1STATCLR = 0x40;// clear the Overflow
    SPI1CON = 0x8720; //SPI ON, 16 bits transfer, SMP=1, Master mode 0x8720
    SPI1CONbits.ON = 0; /* Reset SPI FIFO */
    SPI1CONbits.ON = 1; /* Enable again SPI */
}

```

Obr. 4. 4 Inicializace SPI linky

Simulátor umožňuje simulovat celkem čtyři kanály. Každý z těchto kanálů je *slave*. Pro výběr daného kanálu se používá *Chipselect*. Druhá funkce tedy umožňuje výběr daného kanálu pro zápis dat, což je v tomto případě binární slovo odpovídající hodnotě simulovaného termoelektrického napětí. Vstupem této funkce je tedy definovaný *Chipselect* a datové slovo. Na začátku této funkce dojde opět k resetu celé linky. *Chipselect* vybraného kanálu je nastaven do logické nuly a tím je vybrán příslušný kanál, na který mají být zapsána data. Následuje zápis dat. Aby došlo ke korektnímu zápisu celého datového slova, je zde *while* smyčka, která čeká, dokud data nebudou

zapsána. Jakmile jsou všechna data zapsána, *chipselect* vybraného kanálu se nastaví do logické jedničky. Po tomto kroku je mikroprocesor opět připraven zapsat další data na jiný nebo stejný kanál.

```

void WriteDataWordSPI1(unsigned int definedChipSelect, unsigned short dataWord)
{
    int rData;

    SPI1CONbits.ON = 0; /* Reset SPI FIFO */
    SPI1CONbits.ON = 1; /* Enable again SPI */

    rData = SPI1BUF; // clears the receive buffer
    SPI1STATCLR = 0x40; // clear the Overflow

    switch(definedChipSelect)
    {
        case(1):
            CS1_TC1 = 0; //put port D1 - TC1 to logical 0
            break;
        case(2):
            CS1_TC2 = 0; //put port D2 - TC2 to logical 0
            break;
        case(3):
            CS1_TC3 = 0; //put port D3 - TC3 to logical 0
            break;
        case(4):
            CS1_TC4 = 0; //put port D6 - TC4 to logical 0
            break;
    }

    SPI1BUF = dataWord; // transmit data*/
    while(!SPI1STATbits.SPIRBF); //wait for end transmission*/

    switch(definedChipSelect)
    {
        case(1):
            CS1_TC1 = 1; //put port D1 - TC1 to logical 1
            break;
        case(2):
            CS1_TC2 = 1; //put port D2 - TC2 to logical 1
            break;
        case(3):
            CS1_TC3 = 1; //put port D3 - TC3 to logical 1
            break;
        case(4):
            CS1_TC4 = 1; //put port D6 - TC4 to logical 1
            break;
    }
}

```

Obr. 4. 5 Zápis dat přes SPI linku

4.3.2 Knihovna LCD

Tato knihovna obsahuje, stejně jako knihovna pro SPI, hlavičkový soubor LDC.h, kde jsou deklarovány všechny funkce pro obsluhu displeje a zdrojový kód daných funkcí LCD.c. Jelikož u displeje nejde pouze o zapsání dat, je pro obsluhu displeje mnohem více funkcí, než pro obsluhu SPI linky. Zdrojový kód obsahuje například funkce pro inicializaci displeje, nastavení kontrastu, rolování displeje apod.

Nejdůležitějšími funkcemi jsou ty pro zápis dat na displej. Jelikož je displej ovládaný pomocí SPI linky, knihovna obsahuje funkci pro její inicializaci. Tato funkce je stejná, jako u knihovny SPI, pouze s rozdílným indexem.

Jelikož se nedá na displej zapsat slovo jako celek, ale je potřeba na displej zapisovat po jednotlivých bitech, bylo nutné vytvořit tzv. vyhledávací tabulku (*LookUp Table*). Daný bit odpovídá hodnotě v tabulce, která je v hexadecimální soustavě.

```

static const unsigned char BitReverseTable256[] =
{
    0x00, 0x80, 0x40, 0xC0, 0x20, 0xA0, 0x60, 0xE0, 0x10, 0x90, 0x50, 0xD0, 0x30, 0xB0, 0x70, 0xF0,
    0x08, 0x88, 0x48, 0xC8, 0x28, 0xA8, 0x68, 0xE8, 0x18, 0x98, 0x58, 0xD8, 0x38, 0xB8, 0x78, 0xF8,
    0x04, 0x84, 0x44, 0xC4, 0x24, 0xA4, 0x64, 0xE4, 0x14, 0x94, 0x54, 0xD4, 0x34, 0xB4, 0x74, 0xF4,
    0x0C, 0x8C, 0x4C, 0xCC, 0x2C, 0xAC, 0x6C, 0xEC, 0x1C, 0x9C, 0x5C, 0xDC, 0x3C, 0xBC, 0x7C, 0xFC,
    0x02, 0x82, 0x42, 0xC2, 0x22, 0xA2, 0x62, 0xE2, 0x12, 0x92, 0x52, 0xD2, 0x32, 0xB2, 0x72, 0xF2,
    0x0A, 0x8A, 0x4A, 0xCA, 0x2A, 0xAA, 0x6A, 0xEA, 0x1A, 0x9A, 0x5A, 0xDA, 0x3A, 0xBA, 0x7A, 0xFA,
    0x06, 0x86, 0x46, 0xC6, 0x26, 0xA6, 0x66, 0xE6, 0x16, 0x96, 0x56, 0xD6, 0x36, 0xB6, 0x76, 0xF6,
    0x0E, 0x8E, 0x4E, 0xCE, 0x2E, 0xAE, 0x6E, 0xEE, 0x1E, 0x9E, 0x5E, 0xDE, 0x3E, 0xBE, 0x7E, 0xFE,
    0x01, 0x81, 0x41, 0xC1, 0x21, 0xA1, 0x61, 0xE1, 0x11, 0x91, 0x51, 0xD1, 0x31, 0xB1, 0x71, 0xF1,
    0x09, 0x89, 0x49, 0xC9, 0x29, 0xA9, 0x69, 0xE9, 0x19, 0x99, 0x59, 0xD9, 0x39, 0xB9, 0x79, 0xF9,
    0x05, 0x85, 0x45, 0xC5, 0x25, 0xA5, 0x65, 0xE5, 0x15, 0x95, 0x55, 0xD5, 0x35, 0xB5, 0x75, 0xF5,
    0x0D, 0x8D, 0x4D, 0xCD, 0x2D, 0xAD, 0x6D, 0xED, 0x1D, 0x9D, 0x5D, 0xDD, 0x3D, 0xBD, 0x7D, 0xFD,
    0x03, 0x83, 0x43, 0xC3, 0x23, 0xA3, 0x63, 0xE3, 0x13, 0x93, 0x53, 0xD3, 0x33, 0xB3, 0x73, 0xF3,
    0x0B, 0x8B, 0x4B, 0xCB, 0x2B, 0xAB, 0x6B, 0xEB, 0x1B, 0x9B, 0x5B, 0xDB, 0x3B, 0xBB, 0x7B, 0xFB,
    0x07, 0x87, 0x47, 0xC7, 0x27, 0xA7, 0x67, 0xE7, 0x17, 0x97, 0x57, 0xD7, 0x37, 0xB7, 0x77, 0xF7,
    0x0F, 0x8F, 0x4F, 0xCF, 0x2F, 0xAF, 0x6F, 0xEF, 0x1F, 0x9F, 0x5F, 0xDF, 0x3F, 0xBF, 0x7F, 0xFF
};

```

Obr. 4. 6 Vyhledávací tabulka (LookUp Table)

Zvolený displej má celkem čtyři řádky. Před zápisem dat na displej je proto potřeba vybrat daný řádek. Výběr řádku je proveden pomocí příkazu *switch*. Jako původní je nastaven první řádek.

```

void rowOfLCD(BYTE typeOfView, BYTE numberOfRow)
{
    BYTE data;
    BYTE dataOfRow;

    switch(numberOfRow)
    {
        case 1:
            data = 0x00;
            break;
        case 2:
            data = 0x20;
            break;
        case 3:
            data = 0x40;
            break;
        case 4:
            data = 0x60;
            break;

        default:
            data = 0x00;
            break;
    }
}

```

Obr. 4. 7 Funkce pro výběr řádku na displeji

Jeden řádek displeje má celkem 10 znaků. Pro zápis textového řetězce slouží funkce *writeLCDstring*. Tato funkce obsahuje cyklus *for*, který zajišťuje zápis na displej znak po znaku. Cyklus *for* se provede maximálně 10x. V této funkci je také podmínka, která hlídá překročení počtu znaků na jeden řádek.

```

signed int writeLCDstring(unsigned char dataForLCD[], unsigned int countOfChars)
{
    BYTE charCount = 0;

    if(countOfChars <= 10)
    {
        for(charCount=0; charCount < (countOfChars); charCount++)
        {
            writeRawData(dataForLCD[charCount]);
        }
        return 0;
    }
    else if((sizeof(dataForLCD)) > 10)
        return 1;
}

```

Obr. 4. 8 Zápís textového řetězce

Při načtení jednoho znaku, je zavolána funkce pro zápis surových dat. Tato funkce vytvoří data o velikosti jednoho bytu pro zápis na displej a následně zavolá funkci *SPIdataToLCD*.

```

void writeRawData(BYTE inputData)
{
    unsigned short spiPattern;
    BYTE RS = 0;
    BYTE RW = 0;
    BYTE startByte = 0x00;
    BYTE spiToSend = 0x00;

    RS = 1;
    RW = 0;
    startByte = (0xF8 | ((RW << 2) | (RS << 1)));
    MCU_delayus(10);

    spiPattern = (0x00 | (inputData));
    spiToSend = (BYTE) (spiPattern);
    MCU_delayus(10);

    SPIdataToLCD(startByte, spiToSend);
}

```

Obr. 4. 9 Vytvoření bytu pro zápis přes SPI

Funkce *SPIdataToLCD* následně provede zápis dat přes SPI linku na displej. Jelikož je pouze jeden modul *slave*, a sice LCD displej, je zde také jen jeden *Chipselect*. Z tohoto důvodu je možné *Chipselect* hned nastavit do logické nuly pro zápis.

Jako první krok, je v této funkci, reset SPI linky. Po resetu se pošle startovací byte. Startovací byte se posílá, aby byl displej připraven pro správný příjem dat. Po odeslání startovacího bytu se pošle byte s daty. Opět jsou v této funkci *while* smyčky, které hlídají korektní přenos dat. Jakmile jsou data přeneseny, *Chipselect* se nastaví do logické jedničky a přenos dat je ukončen.

```

void SPIdataToLCD(BYTE startByte, BYTE dataByte)
{
    int rData;
    BYTE firstDataByte = 0x00;
    BYTE secondDataByte = 0x00;
    BYTE swapByte = 0x00;

    MCU_delayus(10);

    swapByte = BitReverseTable256[dataByte];
    secondDataByte = (0xF0 & (swapByte << 4)); //nibble 1
    firstDataByte = (0xF0 & (swapByte)); //nibble 2

    SPI2CONbits.ON = 0; //Reset SPI FIFO
    SPI2CONbits.ON = 1; // Enable again SPI

    rData = SPI2BUF; // clears the receive buffer
    SPI2STATCLR = 0x40; // clear the Overflow
    SPI2CON = 0x8320; //8bit, master
    SPI2CONbits.CKE = 1;
    SPI2CONbits.CKP = 0;

    MCU_delayus(10);

    CS_LCD = 0; //put port G9 - CS_LCD to logical 0
    MCU_delayus(10);

    SPI2STATCLR = 0x40;
    SPI2BUF = startByte; //transmit data
    while(!SPI2STATbits.SPIRBF); //wait for end transmission
    MCU_delayus(1);

    SPI2BUF = firstDataByte; //transmit data
    while(!SPI2STATbits.SPIRBF); //wait for end transmission
    MCU_delayus(1);

    SPI2BUF = secondDataByte; //transmit data
    while(!SPI2STATbits.SPIRBF); //wait for end transmission
    MCU_delayus(5);

    CS_LCD = 1; //put port G9 - CS_LCD to logical 1
    MCU_delayus(100);
}

```

Obr. 4. 10 Zápís dat na displej přes SPI

4.4 Řídicí software

Tato kapitola se zabývá vývojem řídicího softwaru a popisem jednotlivých funkcí. Hlavní zdrojový kód, main.c, obsahuje řídicí kód pro enkodéry, tlačítka a funkce. Na začátku hlavního zdrojového kódu, jsou deklarovány všechny použité knihovny. Tyto knihovny však mají samostatný zdrojový kód. V průběhu programu, jsou tyto knihovny otevřeny pouze na takovou dobu, aby se provedl jejich obsah. Dále jsou zde deklarovány globální proměnné. Globální proměnné je možné použít v rámci celého zdrojového kódu.

4.4.1 Enkodéry

Pro nastavení simulované teploty nebo termoelektrického napětí slouží enkodéry. Na začátku řídicí funkce pro enkodéry jsou deklarovány lokální proměnné. Tyto proměnné mohou být použity pouze pro funkci, ve které jsou deklarovány. Není možné tyto proměnné využívat i v rámci celého zdrojového kódu. Lokální proměnné využívané ve funkci pro enkodéry jsou:

- newvalue (INT)
- value (INT)
- tempData (CHAR)

Dále pak tato funkce využívá tyto globální proměnné:

- floatVal (FLOAT)
- Temp (FLOAT)
- ShowTemp (BOOL)
- oldvalue (INT)

Datový typ INT je určen pro celá čísla a jeho velikost je 8 nebo 16 bitů. Dále se pro celá čísla nebo znaky používá datový typ CHAR o velikosti 8 bitů. Pro racionální čísla, je určen datový typ FLOAT, jehož velikost je 32 bitů. Jako poslední, je použitý datový typ BOOL, který může nabývat pouze dvou stavů, a sice 0 nebo 1.

Po deklaraci lokálních proměnných, se do proměnné *newvalue* zapíše nová hodnota z portu. Následně je tato hodnota umocněna starou hodnotou z předchozího stavu. Výsledek je pak zapsán do proměnné *value*.

Pak jsou zde podmínky, které kontrolují otočení enkodéru. Jestliže došlo k otočení enkodéru směrem doleva, hodnota proměnné Temp nebo *floatVal* se dekrementuje o jednu desetinu. Proměnná Temp obsahuje hodnotu simulované teploty a proměnná floatVal obsahuje hodnotu simulovaného termoelektrického napětí. Pro určení zda se má dekrementovat proměnná Temp nebo *floatVal*, je zde vložena podmínka, která kontroluje, zda je proměnná ShowTemp ve stavu logické jedničky nebo logické nuly. Pokud je *ShowTemp* nastavena do log. jedničky, znamená to, že zobrazovaná veličina na displeji je simulovaná teplota a bude se tedy dekrementovat proměnná Temp. V opačném případě, kdy je *ShowTemp* nastavena do log. nuly, se bude dekrementovat proměnná *floatVal* a zobrazovaná veličina je simulované napětí. Dojde-li k otočení enkodéru směrem doprava, proměnné Temp nebo *floatVal* se inkrementují o jednu desetinu. Opět je zde podmínka, která určuje, jaká proměnná se má inkrementovat. Při inkrementaci nebo dekrementaci proměnné *floatVal* je zde ještě podmínka pro omezení rozsahu simulovaného napětí. Simulované napětí je omezeno na $\pm 60\text{mV}$.

Jakmile je hodnota proměnné inkrementována nebo dekrementována, následuje formátování textového řetězce pro zobrazení na displej. Jelikož je textový řetězec pro zobrazení teploty jiný, než pro zobrazení napětí, je zde opět podmínka, která kontroluje stav proměnné *ShowTemp*. Simulátor umožňuje simulovat kladné i záporné hodnoty napětí a teploty. Z tohoto důvodu jsou dva textové řetězce pro zobrazení teploty a dva pro zobrazení napětí. Pro zobrazení teploty jsou textové řetězce v těchto tvarech:

- textový řetězec pro kladnou hodnotu teploty
 - "TC1_+%06.1fC\r"
- textový řetězec pro zápornou hodnotu teploty
 - "TC1_%07.1fC\r"

Pro zobrazení simulovaného napětí jsou textové řetězce v tomto tvaru:

- textový řetězec pro kladnou hodnotu napětí
 - "TC1_+%.1fmV\r"
- textový řetězec pro zápornou hodnotu napětí
 - "TC1_%.1fmV\r"

Rozdíl pro zobrazení záporné nebo kladné hodnoty je pouze v počtu zobrazovaných znaků před desetinnou tečkou. Při zobrazování záporné hodnoty, se znaménko mínus přidá automaticky a je počítáno do definovaného počtu zobrazovaných znaků před desetinnou tečkou. U kladné hodnoty je nutno definovat o jeden znak méně a znaménko plus připsat zvlášť, před definici počtu znaků.

```
void Encoder1(signed int inputValue)
{
    int newvalue = 0;
    int value = 0;
    char tempData[13];

    newvalue = PORTE; // read the current port status
    value = newvalue ^ oldvalue1; // flag each port bit that has changed

    if(value & 0x01)
    {
        if (newvalue & 0x01)
        {
            if (newvalue & 0x02)
            {
                if (ShowTemp == 1)
                    Temp1 = Temp1 - 0.1;
                else if (floatVal1 > -60.0)
                    floatVal1 = floatVal1 - 0.1;
            }

            else if (ShowTemp == 1)
                Temp1 = Temp1 + 0.1;
            else if (floatVal1 < 60.0)
                floatVal1 = floatVal1 + 0.1;

            if ((ShowTemp == 1) && (Temp1 >= 0.0))
                sprintf(tempData, "TC1_+%.1fC\r", Temp1);
            else if ((ShowTemp == 1) && (Temp1 <= 0.0))
                sprintf(tempData, "TC1_%.1fC\r", Temp1);
            else if (floatVal1 >= 0.0)
                sprintf(tempData, "TC1_+%.1fmV\r", floatVal1);
            else if (floatVal1 < 0.0)
                sprintf(tempData, "TC1_%.1fmV\r", floatVal1);

            if (ShowTemp == 1)
                TC_TEMP(tempData);
            else if (ShowTemp == 0)
                TC_VOLTAGE(tempData);

            MCU_delayus(10);
        }
    }
    oldvalue1 = newvalue; // save current port status
}
```

Obr. 4. 11 Řídící funkce pro enkodéry

Naformátovaný textový řetězec je uložen do proměnné *tempData*. Proměnná *tempData* je pole o velikosti 13 znaků. Řetězec pro zobrazení teploty obsahuje 13 znaků, ale řetězec pro zobrazení napětí obsahuje pouze 12 znaků. Při zobrazování napětí je tedy poslední znak pole nulový.

Po naformátování řetězce, je zavolána funkce pro zápis na displej. Je-li zobrazována teplota, volá se funkce `TC_TEMP`. Jestliže je zobrazováno napětí, volá se funkce `TC_VOLTAGE`. O tom, která funkce má být zavolána, rozhoduje podmínka, která kontroluje stav proměnné `ShowTemp`.

Řídící funkce jsou pro všechny čtyři enkodéry stejné, pouze s rozdílným indexem názvu funkce a jednotlivých proměnných.

4.4.2 Funkce `TC_VOLTAGE`

Tato funkce slouží pro zápis simulovaného napětí přes SPI linku, na výstup daného kanálu a zobrazení na displeji. Vstupem této funkce je textový řetězec naformátovaný ve funkci pro obsluhu enkodérů. Funkce `TC_VOLTAGE` je využívána i pro ovládání simulátoru přes USB port. Pomocí aplikace pro sériovou komunikaci je do mikroprocesoru přes USB port poslán textový řetězec stejného formátu, jaký vytvoří funkce pro enkodéry.

Lokální proměnné v této funkci jsou:

- `tempArray` (CHAR)
- `selector` (CHAR)
- `tempLCDdata` (CHAR)
- `dataDAC` (SHORT)
- `TC_select` (INT)
- `valASCIIToFloat` (FLOAT)

Na začátku této funkce jsou podmínky, které kontrolují správný formát textového řetězce. Každý znak řetězce se kontroluje zvlášť. Je-li řetězec ve správném formátu, do pole `tempArray` se zapíše hodnota simulovaného napětí. Velikost pole `tempArray` je 6 znaků. Simulované napětí je zobrazováno na dvě celá čísla a jedno desetinné místo. Včetně desetinné tečky se jedná celkem o 5 znaků, přičemž šestý znak pole je vynulován pomocí příkazu `NULL`, což se v `ASCII` kódu zapíše jako `'\0'`.

Následně je potřeba určit, na který kanál se má dané napětí zapsat. Do pole `selector` se zapíše třetí znak z textového řetězce. Dle formátu textového řetězce, který je zobrazen výše, je patrné, že právě třetí znak označuje daný kanál, na kterém má být simulované napětí nastaveno. Jelikož se však jedná o znak, nikoliv číslici, je potřeba tento znak převést. Pomocí příkazu `atoi` se převede znak z `ASCII` kódu na číslici. Datový typ převedené číslice je pak `integer` (INT). V neposlední řadě je nutné převést znaky v poli `tempArray` na racionální číslo. Příkaz `atof` převede znaky z `ASCII` kódu na racionální číslo, jehož datový typ je `FLOAT`.

```

void TC_VOLTAGE(char bufferFromUSB[])
{
    char tempArray[6];
    char selector[1];
    unsigned char tempLCDdata[10];
    unsigned short dataDAC;
    unsigned int TC_select;
    float valASCIIToFloat = 0.0;

    if((bufferFromUSB[0] == 'T') && (bufferFromUSB[1] == 'C') && (bufferFromUSB[3] == '_' ) && (bufferFromUSB[11] == '\r'))
    {
        if((bufferFromUSB[9] == 'm') && (bufferFromUSB[10] == 'V'))
        {
            tempArray[0] = bufferFromUSB[4];
            tempArray[1] = bufferFromUSB[5];
            tempArray[2] = bufferFromUSB[6];
            tempArray[3] = bufferFromUSB[7];
            tempArray[4] = bufferFromUSB[8];
            tempArray[5] = '\0';

            selector[0] = (bufferFromUSB[2]);
            TC_select = atoi(selector);

            valASCIIToFloat = atof(tempArray);
        }
    }
}

```

Obr. 4. 12 Převod ASCII znaků na čísla - TC_VOLTAGE

Další část funkce TC_VOLTAGE provádí přepočítání racionálního čísla na datové slovo pro DA převodník. Datové slovo je poté přes SPI linku posláno do DA převodníku. Tato část funkce je nesmírně důležitá, protože podle vypočítaného datového slova, je simulované napětí nastavené na výstupu simulátoru. Při chybném výpočtu by pak na výstupu simulátoru bylo nastaveno jiné napětí, než je zobrazeno na displeji. Tato část funkce je pomocí složených podmínek rozdělena do třech bloků. První z nich porovnává hodnotu proměnné *valASCIIToFloat*, která obsahuje hodnotu napětí s nulou. Tato podmínka také kontroluje, zda je proměnná *TC_select* v rozmezí od 1 do 4. Je-li hodnota simulovaného napětí rovna nule a proměnná *TC_select* je v daném rozmezí, je podmínka splněna a na výstup daného kanálu se nastaví napětí 0mV. Jestliže tato podmínka není splněna, přechází se k podmínce druhé.

Druhá podmínka opět kontroluje, jestli je proměnná *TC_select* v daném rozmezí. Dále tato podmínka kontroluje, zda je hodnota simulovaného napětí menší než nula a zároveň větší nebo rovna -60, což je nejmenší možné simulovatelné napětí. Pokud je tato podmínka splněna, provede se přepočítání na datové slovo. Jelikož hodnota simulovaného napětí je v tomto případě záporná, je potřeba pomocí příkazu *fabs* získat absolutní hodnotu. Absolutní hodnota je následně vynásobena číslem 0,0675, což je koeficient napěťového děliče. Výsledek je vydělen hodnotou 4,096, která odpovídá velikosti referenčního napětí. Tato hodnota je pak vynásobena mocninou 2^{15} . Mocnina 2^{15} je polovina celkového počtu kroků DA převodníku. Posledním krokem je odečtení vypočítané hodnoty od datového slova 0x8000, což odpovídá hodnotě napětí 0mV. V případě, že tato podmínka není splněna, program pokračuje k podmínce třetí.

Třetí podmínka kontroluje, zda je hodnota simulovaného napětí větší než nula a zároveň menší nebo rovna 60. Maximální hodnota simulovaného napětí je totiž 60mV. Dále tato podmínka kontroluje proměnnou *TC_select*. Je-li podmínka splněna, provede se tentýž přepočítání, jako v druhé podmínce. V tomto případě však není potřeba určovat absolutní hodnotu nebo zdaje hodnota simulovaného napětí kladná. Výsledek výpočtu se pak přičte k datovému slovu 0x8000. Po přepočtu je následně zavolána funkce *WriteDataWordSPI1*, která pošle datové slovo do DA převodníku.


```

if((valASCIIToFloat == 0.0) && ((TC_select >= 1) && (TC_select <= 4)))
{
    dataDAC = 0x8000;
    WriteDataWordSPI1(TC_select, dataDAC);
}

else if(((TC_select >= 1) && (TC_select <= 4)) && (valASCIIToFloat < 0.0) && (valASCIIToFloat >= -60.0))
{
    dataDAC = (0x8000) - (((fabs(valASCIIToFloat) * (0.0675)) / 4.096)) * (powf(2,15)));
    WriteDataWordSPI1(TC_select, dataDAC);
    MCU_delayus(10);

    putUSBUSART(RESPONSE_P, sizeof(RESPONSE_P));
}

else if(((TC_select >= 1) && (TC_select <= 4)) && (valASCIIToFloat > 0.0) && (valASCIIToFloat <= 60.0))
{
    dataDAC = (0x8000) + (((valASCIIToFloat * (0.0675)) / 4.096)) * (powf(2,15));

    WriteDataWordSPI1(TC_select, dataDAC);
    MCU_delayus(10);

    putUSBUSART(RESPONSE_P, sizeof(RESPONSE_P));
}

```

Obr. 4. 13 *Přepočet napětí na datové slovo - TC_VOLTAGE*

Poslední částí této funkce je zobrazení simulovaného napětí na displeji. Jako první je potřeba zjistit, o který kanál simulátoru se jedná, a tím zároveň určit, na jaký řádek displeje se hodnota simulovaného napětí zapíše. Pro určení daného kanálu slouží v této funkci proměnná *TC_select*. Proměnná *TC_select* může nabývat hodnot v rozmezí 1 až 4. Na základě proměnné *TC_select* je pomocí příkazu *switch* vybrána podfunkce pro příslušný řádek displeje. V této podfunkci je zavolána funkce *rowOfLCD*, která nastaví kurzor displeje na vybraný řádek a první znak zleva. Pomocí podmínky, která porovnává hodnotu zobrazovaného napětí, se vybere správný formát textového řetězce. Textové řetězce se i v tomto případě rozlišují podle toho, je-li hodnota simulovaného napětí záporná nebo kladná.

```

switch(TC_select)
{
    case 1:
        rowOfLCD(0,1);
        if(valASCIIToFloat >= 0.0)
            sprintf(tempLCDdata, "TC1+%04.1fmV", valASCIIToFloat);
        else
            sprintf(tempLCDdata, "TC1%05.1fmV", valASCIIToFloat);
        break;

    case 2:
        rowOfLCD(0,2);
        if(valASCIIToFloat >= 0.0)
            sprintf(tempLCDdata, "TC2+%04.1fmV", valASCIIToFloat);
        else
            sprintf(tempLCDdata, "TC2%05.1fmV", valASCIIToFloat);
        break;

    case 3:
        rowOfLCD(0,3);
        if(valASCIIToFloat >= 0.0)
            sprintf(tempLCDdata, "TC3+%04.1fmV", valASCIIToFloat);
        else
            sprintf(tempLCDdata, "TC3%05.1fmV", valASCIIToFloat);
        break;

    case 4:
        rowOfLCD(0,4);
        if(valASCIIToFloat >= 0.0)
            sprintf(tempLCDdata, "TC4+%04.1fmV", valASCIIToFloat);
        else
            sprintf(tempLCDdata, "TC4%05.1fmV", valASCIIToFloat);
        break;
}
writeLCDstring(tempLCDdata, sizeof(tempLCDdata));

```

Obr. 4. 14 *Zobrazení simulovaného napětí na displeji*

4.4.3 Funkce TC_TEMP

Funkce *TC_TEMP* zajišťuje v první řadě přepočítání simulované teploty na termoelektrické napětí. Přepočítané napětí je následně nastaveno na výstup daného kanálu. Dále je pomocí této funkce simulovaná teplota zobrazena na displeji. Tato funkce je obdobou funkce *TC_VOLTAGE*.

Na začátku funkce jsou deklarovány lokální proměnné, které jsou:

- tempArray (CHAR)
- selector (CHAR)
- tempLCDdata (CHAR)
- dataDAC (SHORT)
- TC_select (INT)
- TempVal (FLOAT)
- recalTempVal (FLOAT)

Stejně jako u funkce *TC_VOLTAGE*, je i zde vstupem funkce textový řetězec. Textový řetězec obsahuje 12 znaků. Po deklarování lokálních proměnných následuje podmínka, která kontroluje, zda je textový řetězec správně naformátován. Při splnění podmínky jsou jednotlivé číselné znaky zapsány do pole *tempArray*. Jakmile je pole *tempArray* vytvořeno, je potřeba tyto znaky v ASCII kódu převést na číslíce. Pomocí funkce *atof* se převedou znaky z pole *tempArray* na číslo, jehož hodnota odpovídá simulované teplotě. Hodnota simulované teploty je uložena do proměnné *TempVal*. Jelikož hodnota simulované teploty je racionální číslo, datový typ této proměnné je *FLOAT*. Dále je potřeba získat index daného kanálu, na který se má simulovaná teplota nastavit. V pořadí třetí znak textového řetězce obsahuje daný index. Jelikož se jedná opět o znak v ASCII kódu, je potřeba jej také převést. Převod znaku na číslíci je proveden pomocí příkazu *atoi*. Následně je tato číslíci zapsána do proměnné *TC_select*.

```
void TC_TEMP(char bufferFromUSB[])
{
    char tempArray[7];
    char selector[1];
    unsigned char tempLCDdata[10];
    unsigned short dataDAC;
    unsigned int TC_select;
    float TempVal = 0.0;
    float recalTempVal = 0.0;

    if((bufferFromUSB[0] == 'T') && (bufferFromUSB[1] == 'C') && (bufferFromUSB[3] == '_') && (bufferFromUSB[12] == '\r'))
    {
        if(bufferFromUSB[11] == 'C')
        {
            tempArray[0] = bufferFromUSB[4];
            tempArray[1] = bufferFromUSB[5];
            tempArray[2] = bufferFromUSB[6];
            tempArray[3] = bufferFromUSB[7];
            tempArray[4] = bufferFromUSB[8];
            tempArray[5] = bufferFromUSB[9];
            tempArray[6] = bufferFromUSB[10];
            tempArray[7] = '\0';

            selector[0] = (bufferFromUSB[2]);
            TC_select = atoi(selector);
            TempVal = atof(tempArray);
        }
    }
}
```

Obr. 4. 15 Převod ASCII znaků na čísla - TC_TEMP

Po převedení znaků byla získána hodnota simulované teploty. Dané teplotě odpovídá jisté termoelektrické napětí. Hodnotu simulované teploty je proto potřeba přepočítat na příslušné termoelektrické napětí. V tabulkovém editoru MS Excel byla vytvořena tabulka, která obsahuje teplotu teplého konce (*Hot Junction*) termočlánu a k ní příslušné referenční termoelektrické napětí dle katalogového listu termočlánu. Teplota studeného konce (*Cold Junction*) je 0°C. Jelikož simulátor umožňuje simulovat pouze jeden typ termočlánu, byl pro tuto práci vybrán typ N.

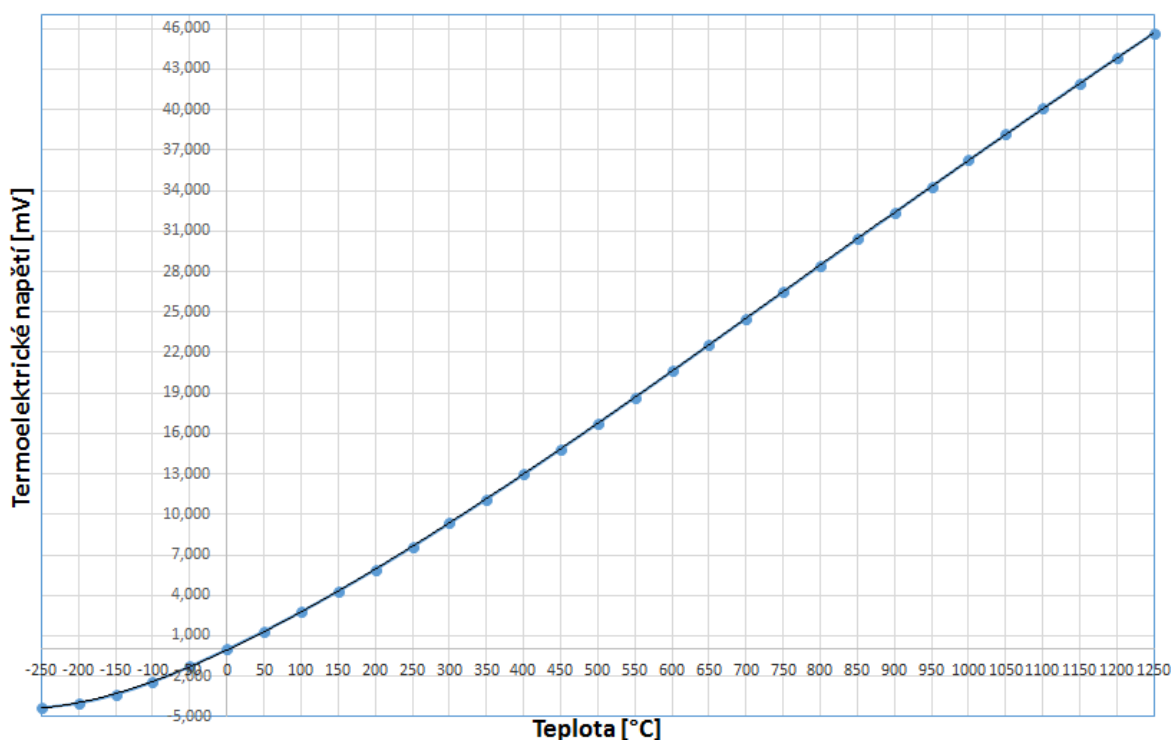
Tab. 4. 1 Referenční tabulka

Teplota [°C]	Referenční napětí [mV]	Vypočítané napětí [mV]	Odchylka Δ [mV]
-250	-4,31	-4,36	0,04
-200	-3,99	-3,95	-0,04
-150	-3,34	-3,28	-0,06
-100	-2,41	-2,38	-0,03
-50	-1,27	-1,30	0,03
0	0,00	-0,06	0,06
50	1,34	1,30	0,04
100	2,77	2,76	0,01
150	4,30	4,31	-0,01
200	5,91	5,94	-0,03
250	7,60	7,63	-0,03
300	9,34	9,37	-0,03
350	11,14	11,15	-0,02
400	12,97	12,98	-0,01
450	14,85	14,84	0,00
500	16,75	16,74	0,01
550	18,67	18,65	0,02
600	20,61	20,59	0,02
650	22,57	22,55	0,02
700	24,53	24,52	0,01
750	26,49	26,49	0,00
800	28,46	28,46	-0,01
850	30,42	30,43	-0,01
900	32,37	32,39	-0,02
950	34,32	34,34	-0,02
1000	36,26	36,27	-0,01
1050	38,18	38,18	0,00
1100	40,09	40,08	0,01
1150	41,98	41,96	0,02
1200	43,85	43,84	0,01
1250	45,69	45,72	-0,03

Dále tato tabulka obsahuje vypočítané napětí a odchylku vypočítané hodnoty. Odchylka byla vypočítána jako absolutní hodnota rozdílu referenčního napětí a vypočítaného napětí. Maximální odchylka je -0,06mV.

Teplota v referenční tabulce je v rozmezí od -250°C do +1250°C s pevným krokem 50°C. Z těchto referenčních hodnot je sestaven bodový graf (Obr. 4. 16), jehož body jsou proloženy regresní křivkou. Jedná se o polynomicou regresi, jejíž výslednou rovnicí je polynom šestého řádu. Pomocí výsledného polynomu je následně vypočítáno termoelektrické napětí, kde za proměnnou x je dosazena hodnota teploty. Výsledný polynom je v tomto tvaru:

$$y = 6,499 \cdot 10^{-18}x^6 - 2,344 \cdot 10^{-14}x^5 + 3,390 \cdot 10^{-11}x^4 - 3,166 \cdot 10^{-8}x^3 + 2,499 \cdot 10^{-5}x^2 + 2,604 \cdot 10^{-2}x - 6,200 \cdot 10^{-2} \quad (4.1.)$$



Obr. 4. 16 Bodový graf s regresní křivkou

Přepočítaná hodnota termoelektrického napětí je následně uložena do proměnné *recalcTempVal*. Jakmile je hotový přepočet z teploty na napětí, pokračuje funkce výpočtem datového slova pro DA převodník. Obdobně jako u funkce *TC_VOLTAGE* jsou zde podmínky, které rozdělují tuto část funkce na tři bloky. První podmínka porovnává hodnotu přepočítaného napětí s nulou. Jestliže je tato podmínka splněna, datové slovo pro DA převodník je 0x8000, což odpovídá napětí o velikosti 0mV. Druhá podmínka kontroluje, jestli je hodnota napětí menší než nula a zároveň větší než -60, což je nejmenší možné simulované napětí. Poslední podmínka kontroluje, jestli je hodnota napětí větší než nula a zároveň menší než +60, což je maximální simulované napětí. Výpočet datového slova je stejný jako ve funkci *TC_VOLTAGE*.

```

if((recalcTempVal == 0.0) && ((TC_select >= 1) && (TC_select <= 4)))
{
    dataDAC = 0x8000;
    WriteDataWordSPI1(TC_select, dataDAC);
}

else if(((TC_select >= 1) && (TC_select <= 4)) && (recalcTempVal < 0.0) && (recalcTempVal >= -60.0))
{
    dataDAC = (0x8000) - (((fabs(recalcTempVal) * (0.0675)) / 4.096)) * (powf(2,15));
    WriteDataWordSPI1(TC_select, dataDAC);
    MCU_delayus(10);

    putUSBUSART(RESPONSE_P, sizeof(RESPONSE_P));
}

else if(((TC_select >= 1) && (TC_select <= 4)) && (recalcTempVal > 0.0) && (recalcTempVal <= 60.0))
{
    dataDAC = (0x8000) + (((recalcTempVal * (0.0675)) / 4.096)) * (powf(2,15));

    WriteDataWordSPI1(TC_select, dataDAC);
    MCU_delayus(10);

    putUSBUSART(RESPONSE_P, sizeof(RESPONSE_P));
}

```

Obr. 4. 17 *Přepočítání napětí na datové slovo - TC_TEMP*

Poslední část funkce TC_TEMP zajišťuje zobrazení simulované teploty na displeji. Pomocí hodnoty proměnné TC_select, která určuje index kanálu a příkazu switch se vybere příslušný řádek na displeji. Po nastavení řádku jsou zde podmínky, které určují formát textového řetězce. Formát se určí podle toho, je-li hodnota zobrazované teploty kladná nebo záporná. Následně je do textového řetězce zapsána hodnota simulované teploty a je zavolána funkce writeLCDstring, která zapíše data přes SPI linku na displej.

```

switch(TC_select)
{
    case 1:
        rowOfLCD(0,1);
        if(TempVal >= 0.0)
            sprintf(tempLCDdata, "T1+%06.1fC", TempVal);
        else
            sprintf(tempLCDdata, "T1%07.1fC", TempVal);
        break;

    case 2:
        rowOfLCD(0,2);
        if(TempVal >= 0.0)
            sprintf(tempLCDdata, "T2+%06.1fC", TempVal);
        else
            sprintf(tempLCDdata, "T2%07.1fC", TempVal);
        break;

    case 3:
        rowOfLCD(0,3);
        if(TempVal >= 0.0)
            sprintf(tempLCDdata, "T3+%06.1fC", TempVal);
        else
            sprintf(tempLCDdata, "T3%07.1fC", TempVal);
        break;

    case 4:
        rowOfLCD(0,4);
        if(TempVal >= 0.0)
            sprintf(tempLCDdata, "T4+%06.1fC", TempVal);
        else
            sprintf(tempLCDdata, "T4%07.1fC", TempVal);
        break;
}

writeLCDstring(tempLCDdata, sizeof(tempLCDdata));

```

Obr. 4. 18 *Zobrazení simulované teploty na displeji*

4.4.4 Přepínání zobrazení simulované veličiny

Další řídicí funkcí v hlavním zdrojovém kódu je funkce pro přepínání zobrazení simulované veličiny, tedy napětí a teploty. K přepnutí slouží mikrospínač, který je umístěn na DPS. Řídicí funkce je umístěna v hlavní *while* smyčce a využívá tyto proměnné:

- SwitchTemp (BOOL)
- ShowTemp (BOOL)
- Temp₁₋₄ (FLOAT)
- floatVal₁₋₄ (FLOAT)

Mikrospínač je v rozepnutém stavu nastaven do logické jedničky. Podmínka porovnává stav portu mikroprocesoru, ke kterému je mikrospínač připojen s logickou jedničkou. Je-li tato podmínka splněna, proměnná *SwitchTemp* je nastavena do logické nuly.

Jakmile dojde ke stisku, dojde k uzemnění mikrospínače a tedy k přechodu do stavu logické nuly. Následuje druhá podmínka, která porovnává stav portu s logickou nulou. Při splnění této podmínky následuje další podmínka, která kontroluje, jaká veličina je právě zobrazena na displeji a zároveň, je-li proměnná *SwitchTemp* v logické nule. To znamená, že mikrospínač byl od posledního stisku uvolněn. Tato podmínka zabraňuje náhodnému přepínání simulované veličiny při trvalém stisku mikrospínače. Informaci o aktuální veličině zobrazované na displeji nám udává stav proměnné *ShowTemp*. Jestliže je proměnná *ShowTemp* v logické jedničce, aktuálně zobrazovaná veličina je simulovaná teplota, v opačném případě je na displeji zobrazeno simulované napětí.

Jestliže je aktuálně na displeji zobrazená simulovaná teplota, při stisku tlačítka dojde k přepnutí na zobrazení simulovaného napětí. Globální proměnné floatVal₁₋₄ obsahující hodnoty simulovaného napětí jsou vynulovány. Při přepnutí veličiny z teploty na napětí dojde k nastavení napětí o velikosti 0mV na všechny čtyři kanály. Následně je proměnná *ShowTemp* nastavena do logické nuly a proměnná *SwitchTemp* do logické jedničky. Jelikož data na displej nejdou zapsat všechny najednou, ale je potřeba je zapisovat zvlášť po jednotlivém řádku, je zde pro zápis využitý cyklus *for*. Cyklus *for* proběhne celkově 4x a tím je zajištěn zápis na každý řádek displeje zvlášť. V cyklu *for* je volána funkce *TC_VOLTAGE*, která zajistí nejen zobrazení simulovaného napětí na displej, ale také nastavení simulovaného napětí na výstup simulátoru.

```

if ((PORTBbits.RB2) == 1)
{
    SwitchTemp = 0;
    MCU_delayus(1000);
}

else if((PORTBbits.RB2) == 0)
{
    if ((ShowTemp == 0) && (SwitchTemp == 0))
    {
        ShowTemp = 1;
        SwitchTemp = 1;
        Temp1 = 0.0;
        Temp2 = 0.0;
        Temp3 = 0.0;
        Temp4 = 0.0;

        for (i = 0; i <= 3; i++)
        {
            if (i == 0)
            {
                TC_TEMP ("TC1_+0000.0C\r");
                MCU_delayus(100);
            }
            else if (i == 1)
            {
                TC_TEMP ("TC2_+0000.0C\r");
                MCU_delayus(100);
            }
            else if (i == 2)
            {
                TC_TEMP ("TC3_+0000.0C\r");
                MCU_delayus(100);
            }
            else
            {
                TC_TEMP ("TC4_+0000.0C\r");
                MCU_delayus(100);
            }
        }

        MCU_delayus(1000);
    }

    else if ((ShowTemp == 1) && (SwitchTemp == 0))
    {
        ShowTemp = 0;
        SwitchTemp = 1;
        floatVal1 = 0.0;
        floatVal2 = 0.0;
        floatVal3 = 0.0;
        floatVal4 = 0.0;

        for (i = 0; i <= 3; i++)
        {
            if (i == 0)
            {
                TC_VOLTAGE ("TC1_+00.0mV\r");
                MCU_delayus(100);
            }
            else if (i == 1)
            {
                TC_VOLTAGE ("TC2_+00.0mV\r");
                MCU_delayus(100);
            }
            else if (i == 2)
            {
                TC_VOLTAGE ("TC3_+00.0mV\r");
                MCU_delayus(100);
            }
            else
            {
                TC_VOLTAGE ("TC4_+00.0mV\r");
                MCU_delayus(100);
            }
        }

        MCU_delayus(1000);
    }
}

```

Obr. 4. 19 Řídící funkce pro přepínání zobrazení simulované veličiny

Je-li zobrazovaná veličina napětí a dojde-li k přepnutí na teplotu, proměnné Temp₁₋₄ obsahující hodnoty simulované teploty, se vynulují. Dále jsou proměnné *ShowTemp* a *SwitchTemp* nastaveny do logické jedničky. Stejně jako v předchozím případě, je pro zápis dat na displej a nastavení nulového napětí na výstup, použitý cyklus *for*. V tomto případě je však volána funkce *TC_TEMP*.

5 Testování a verifikace

Tato kapitola se zabývá konečným testováním simulátoru a následnou verifikací výsledků. V rámci testování bylo provedeno měření. Měření probíhalo pro každý kanál simulátoru zvlášť. Cílem měření bylo na každém kanálu naměřit skutečnou hodnotu simulovaného napětí. Simulované napětí bylo měřeno v maximálním rozsahu, který simulátor umožňuje, tedy od -60mV do +60mV. Při simulování teploty, bylo opět cílem naměřit skutečnou simulovanou hodnotu napětí, odpovídající simulované teplotě. Teplota byla simulována dle katalogového listu termočlánu typu N, tedy od -250°C do +1250°C. Simulátor simuluje teplotu teplého konce (*Hot Junction*). Teplota studeného konce (*Cold Junction*) je 0°C.

Pro měření byl použit měřicí přístroj *Agilent U1401B*. Tento multifunkční měřicí přístroj s kalibrátorem je určen jak pro měření různých signálů, tak pro generování stejnosměrného napětí a jeho pulzů. Multimetr obsahuje například duální LCD displej, funkci HOLD, zaznamenávání dat pro počítač a další. Pracovní teplota přístroje se pohybuje v rozmezí od 0°C do 40°C.

Důležité pro tuto práci jsou však vlastnosti přístroje týkající se měření napětí v řádech milivoltů. Přístroj umožňuje nastavení několika rozsahů pro měření voltů. Nejvhodnější rozsah pro měření milivoltů je $\pm 50\text{mV}$. Při tomto rozsahu je přístroj schopen měřit s rozlišením $1\mu\text{V}$ a přesností $\pm (0,05\% + 5\text{dig.})$. Nominální hodnota vstupní impedance je pak $1\text{G}\Omega$. [14]

Přesnost přístroje je zaručena výrobcem pouze při dodržení určitých podmínek při měření. Při měření je nutné dodržet teplotu prostředí 23°C , $\pm 5^\circ\text{C}$ a vlhkost vzduchu menší než 80%. Při nesplnění těchto podmínek není zaručena daná přesnost přístroje. [14]

Simulátor umožňuje simulovat termoelektrické napětí s přesností na desetiny milivoltu. Pro měření a testování tohoto simulátoru je tak přístroj *Agilent U1401B* naprosto dostačující a přesný.



Obr. 5. 1 Použitý měřicí přístroj [14]

Při měření simulovaného napětí bylo v dané rozsahu naměřeno celkem 25 hodnot. Jelikož simulátor umožňuje simulovat termoelektrické napětí s přesností na jednu desetinu milivoltu, nebyl zvolen pevný krok. Velikost kroku se pohybovala od 4,3mV do 5,7mV. Jakmile bylo provedeno měření na všech čtyřech kanálech simulátoru, byla vypočtena odchylka. Tato odchylka udává rozdíl mezi nastaveným napětím na simulátoru a skutečným naměřeným napětím na daném kanálu. Velikost odchylky se pohybovala v řádech setin milivoltů. Konkrétní naměřené hodnoty simulovaného napětí, včetně odchylky, jsou zobrazeny v tabulce níže. (Tab. 5. 1) Během měření byl také zjištěn rozdíl mezi jednotlivými kanály simulátoru. Jelikož se jedná o rozdíly v řádech setin milivoltu, mohou být tyto odchylky způsobeny jednotlivými spoji na DPS.

Tab. 5. 1 Naměřené hodnoty a jejich odchylky – simulace napětí

U_{SIM} [mV]	TC1 [mV]	TC2 [mV]	TC3 [mV]	TC4 [mV]	$\Delta TC1$ [mV]	$\Delta TC2$ [mV]	$\Delta TC3$ [mV]	$\Delta TC4$ [mV]
-60,0	-59,991	-59,972	-59,971	-59,952	-0,009	-0,028	-0,029	-0,048
-55,7	-55,683	-55,671	-55,672	-55,655	-0,017	-0,029	-0,028	-0,045
-50,7	-50,683	-50,672	-50,682	-50,664	-0,017	-0,028	-0,018	-0,036
-45,7	-45,682	-45,673	-45,683	-45,674	-0,018	-0,027	-0,017	-0,026
-40,7	-40,684	-40,673	-40,684	-40,672	-0,016	-0,027	-0,016	-0,028
-35,7	-35,682	-35,671	-35,682	-35,671	-0,018	-0,029	-0,018	-0,029
-30,7	-30,681	-30,672	-30,673	-30,671	0,002	-0,007	-0,006	-0,008
-25,7	-25,683	-25,672	-25,681	-25,672	-0,017	-0,028	-0,019	-0,028
-20,7	-20,682	-20,673	-20,682	-20,683	-0,018	-0,027	-0,018	-0,017
-15,6	-15,574	-15,573	-15,582	-15,573	-0,026	-0,027	-0,018	-0,027
-10,6	-10,585	-10,574	-10,583	-10,585	-0,015	-0,026	-0,017	-0,015
-5,6	-5,570	-5,572	-5,583	-5,584	-0,030	-0,028	-0,017	-0,016
-0,6	-0,571	-0,573	-0,581	-0,582	-0,029	-0,027	-0,019	-0,018
4,4	4,431	4,432	4,442	4,423	-0,031	-0,032	-0,042	-0,023
9,5	9,542	9,541	9,531	9,522	-0,042	-0,041	-0,031	-0,022
14,5	14,553	14,541	14,532	14,520	-0,053	-0,041	-0,032	-0,020
19,5	19,552	19,543	19,543	19,511	-0,052	-0,043	-0,043	-0,011
24,4	24,454	24,444	24,441	24,422	-0,054	-0,044	-0,041	-0,022
29,4	29,454	29,443	29,432	29,422	-0,054	-0,043	-0,032	-0,022
34,4	34,461	34,452	34,454	34,423	-0,061	-0,052	-0,054	-0,023
39,4	39,462	39,442	39,443	39,422	-0,062	-0,042	-0,043	-0,022
44,3	44,361	44,351	44,342	44,314	-0,061	-0,051	-0,042	-0,014
49,3	49,373	49,352	49,341	49,323	-0,073	-0,052	-0,041	-0,023
54,3	54,363	54,343	54,343	54,321	-0,063	-0,043	-0,043	-0,021
60,0	60,071	60,062	60,042	60,025	-0,071	-0,062	-0,042	-0,025

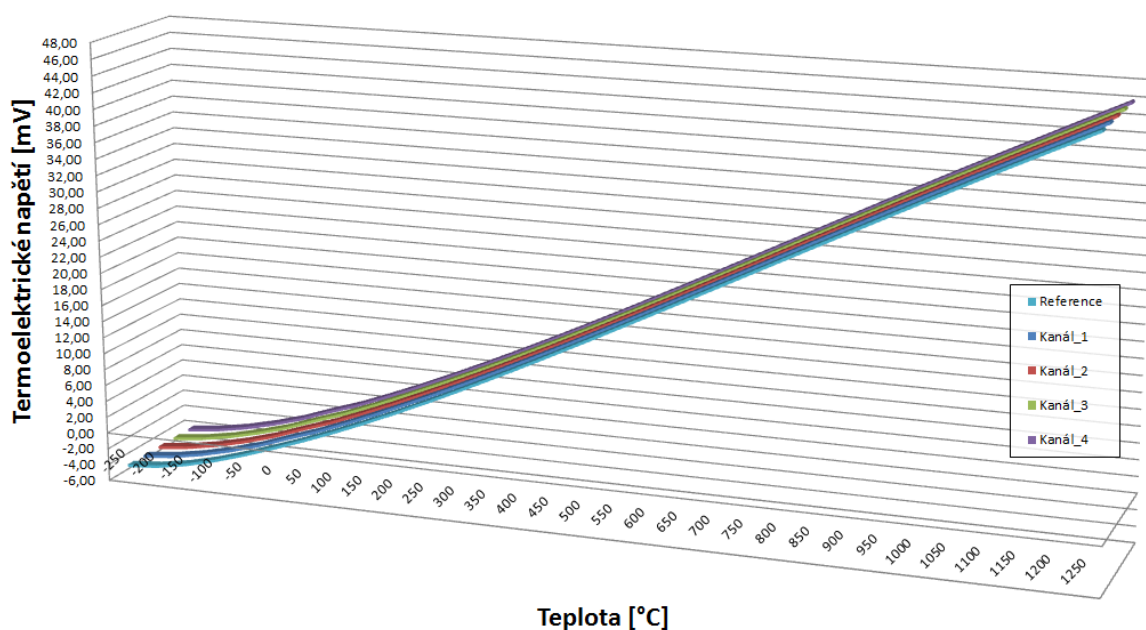
Druhým bodem měření a testování simulátoru, bylo simulovat teplotu termočlánku a na výstupech jednotlivých kanálů simulátoru změřit termoelektrické napětí. Naměřené termoelektrické napětí odpovídá dané teplotě, která je nastavena na simulátoru. V rozsahu od -250°C do 1250°C bylo simulováno celkem 31 hodnot, což odpovídá velikosti kroku 50°C.

Tab. 5. 2 Naměřené hodnoty a jejich odchylky – simulace teploty

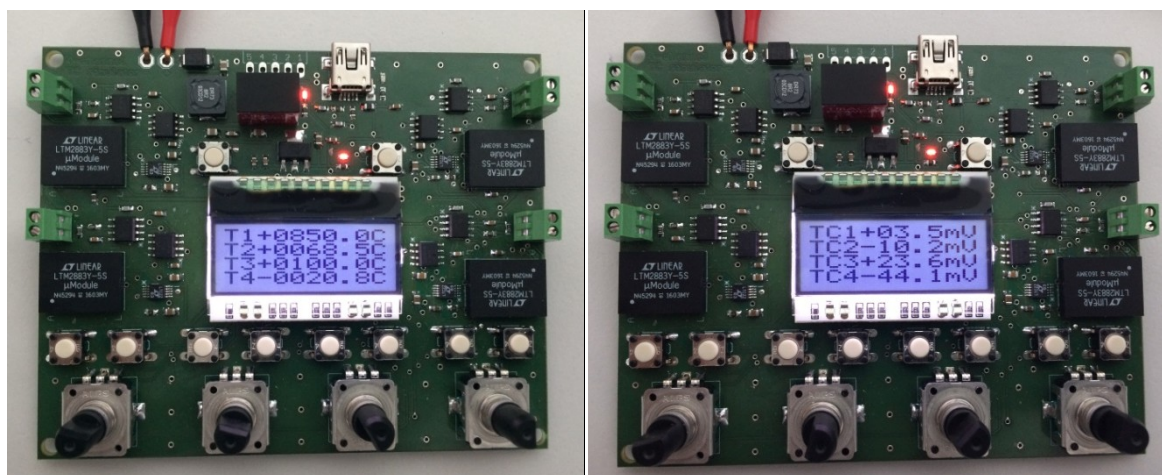
T_{SIM} [°C]	U_{REF} [mV]	TC1 [mV]	TC2 [mV]	TC3 [mV]	TC4 [mV]	$\Delta TC1$ [mV]	$\Delta TC2$ [mV]	$\Delta TC3$ [mV]	$\Delta TC4$ [mV]
-250	-4,31	-4,321	-4,321	-4,322	-4,331	0,008	0,008	0,009	0,018
-200	-3,99	-3,912	-3,910	-3,912	-3,921	-0,078	-0,080	-0,078	-0,069
-150	-3,34	-3,242	-3,241	-3,241	-3,252	-0,094	-0,095	-0,095	-0,084
-100	-2,41	-2,341	-2,342	-2,340	-2,352	-0,066	-0,065	-0,067	-0,055
-50	-1,27	-1,260	-1,263	-1,271	-1,273	-0,009	-0,006	0,002	0,004
0	0,00	0,073	0,071	0,081	0,081	-0,073	-0,071	-0,081	-0,081
50	1,34	1,341	1,343	1,332	1,332	-0,001	-0,003	0,008	0,008
100	2,77	2,802	2,802	2,791	2,803	-0,028	-0,028	-0,017	-0,029
150	4,30	4,352	4,351	4,352	4,344	-0,050	-0,049	-0,050	-0,042
200	5,91	5,981	5,982	5,982	5,972	-0,068	-0,069	-0,069	-0,059
250	7,60	7,671	7,671	7,673	7,663	-0,074	-0,074	-0,076	-0,066
300	9,34	9,413	9,412	9,410	9,401	-0,072	-0,071	-0,069	-0,060
350	11,14	11,202	11,203	11,201	11,192	-0,066	-0,067	-0,065	-0,056
400	12,97	13,031	13,031	13,031	13,024	-0,057	-0,057	-0,057	-0,050
450	14,85	14,891	14,892	14,891	14,882	-0,045	-0,046	-0,045	-0,036
500	16,75	16,792	16,790	16,792	16,781	-0,044	-0,042	-0,044	-0,033
550	18,67	18,714	18,711	18,713	18,713	-0,042	-0,039	-0,041	-0,041
600	20,61	20,653	20,651	20,652	20,642	-0,040	-0,038	-0,039	-0,029
650	22,57	22,611	22,612	22,612	22,602	-0,045	-0,046	-0,046	-0,036
700	24,53	24,571	24,572	24,571	24,561	-0,044	-0,045	-0,044	-0,034
750	26,49	26,542	26,542	26,541	26,530	-0,051	-0,051	-0,050	-0,039
800	28,46	28,523	28,523	28,522	28,513	-0,068	-0,068	-0,067	-0,058
850	30,42	30,491	30,493	30,493	30,471	-0,075	-0,077	-0,077	-0,055
900	32,37	32,450	32,452	32,452	32,442	-0,079	-0,081	-0,081	-0,071
950	34,32	34,401	34,401	34,401	34,391	-0,082	-0,082	-0,082	-0,072
1000	36,26	36,332	36,331	36,332	36,332	-0,076	-0,075	-0,076	-0,076
1050	38,18	38,242	38,243	38,242	38,242	-0,063	-0,064	-0,063	-0,063
1100	40,09	40,141	40,142	40,143	40,132	-0,054	-0,055	-0,056	-0,045
1150	41,98	42,032	42,030	42,031	42,023	-0,056	-0,054	-0,055	-0,047
1200	43,85	43,901	43,902	43,900	43,901	-0,055	-0,056	-0,054	-0,055
1250	45,69	45,793	45,791	45,791	45,782	-0,099	-0,097	-0,097	-0,088

V tabulkovém editoru byla vytvořena tabulka (Tab. 5. 2), která obsahuje referenční napětí pro danou teplotu dle katalogového listu termočlánu typu N. Tabulka dále zaznamenává naměřené hodnoty napětí a odchylky, které udávají rozdíl mezi referenčním a naměřeným napětím.

Jakmile byly všechny naměřené hodnoty zapsány do tabulky, byl z těchto hodnot vytvořen graf. Tento graf zobrazuje závislost termoelektrického napětí na simulované teplotě. Graf obsahuje celkem 5 křivek. Jedna křivka reprezentuje referenční napětí dle katalogového listu termočlánu a zbylé 4 křivky reprezentují jednotlivé kanály simulátoru. Jelikož rozdíly mezi jednotlivými kanály nebyly téměř žádné a křivky byly v zákrytu, byl vytvořen 3D graf.



Obr. 5. 2 Grafické porovnání výsledků měření



Obr. 5. 3 Simulátor během testování

6 Závěr

Cílem této práce bylo navrhnout a vytvořit zařízení pro simulaci termoelektrických článků. Simulátor umožňuje simulovat celkem čtyři termoelektrické články stejného typu nezávisle na sobě. Zařízení obsahuje enkodéry pro nastavení simulované teploty nebo napětí a LCD displej pro zobrazení simulované veličiny.

Ve vývojovém prostředí EAGLE byla navržena deska plošných spojů. Následně byla deska osazena a oživena. Jakmile byla deska oživená a plně funkční, bylo potřeba vyvinout řídicí software a implementovat jej. Software byl vyvíjen ve vývojovém prostředí MPLAB, společnosti Microchip. Na začátku byly vytvořeny knihovny pro komunikaci přes SPI linku a obsluhu LCD displeje. Následně byly vyvinuty řídicí funkce pro obsluhu enkodérů a mikrosplínačů. Jako poslední byly vytvořeny funkce pro simulování teploty a termoelektrického napětí. V tabulkovém editoru MS Excel byla vytvořena referenční tabulka, ze které byl sestrojen bodový graf. Jednotlivé body byly proloženy regresní křivkou. Regresní křivka je polynomického tvaru a šestého řádu. Výsledný polynom regresní křivky byl použit pro přepočítání simulované teploty na termoelektrické napětí.

Pro ověření přesnosti a funkčnosti simulátoru bylo provedeno měření. Měření bylo rozděleno do dvou částí. V první části bylo simulováno termoelektrické napětí. Termoelektrické napětí bylo simulováno v maximálním rozsahu, který simulátor umožňuje, což je od -60mV do $+60\text{mV}$. Měření simulovaného napětí bylo provedeno na všech kanálech zařízení. Celkem bylo na každém kanálu naměřeno 25 různých hodnot. Naměřené hodnoty byly zapsány do tabulky (Tab. 5. 1) a následně byla vypočtena odchylka. Odchylka udává rozdíl mezi naměřeným a nastaveným napětím. Jednotlivé kanály dosahovaly odchylky v řádech setin milivoltu. Maximální odchylka byla $-0,073\text{mV}$. Konkrétní odchylky pro všechny kanály jsou zobrazeny v tabulce (Tab. 5. 1). Při uvážení, že simulátor je schopen simulovat napětí s přesností na desetiny milivoltu, je odchylka v řádech setin milivoltu zanedbatelná.

Druhou částí měření bylo simulování teploty. Teplota byla simulovaná v rozsahu podle katalogového listu termočlánku. Simulátor umožňuje simulovat pouze jeden typ termočlánku. Konkrétně byl pro tuto práci zvolen termočlánek typu N. Simulace tedy probíhala v teplotním rozsahu od -200°C do $+1250^\circ\text{C}$ s pevným krokem 50°C . Naměřené hodnoty byly zapsány do tabulky a následně byl z naměřených hodnot sestaven graf. Stejně jako v předchozím případě byla vypočtena odchylka, která je rovněž zapsána v tabulce z měření (Tab. 5. 2). Odchylky byly v tomto případě vyšší než při simulaci napětí. Maximální odchylka byla v tomto případě $-0,099\text{mV}$. Konkrétní odchylky jsou zobrazeny v tabulce (Tab. 5. 2). Vyšší odchylky jsou způsobeny nepřesností přepočtu simulované teploty na termoelektrické napětí. K odchylce jednotlivého kanálu se v tomto případě ještě přičte odchylka způsobená přepočtem teploty na napětí. Maximální odchylka přepočtu je $-0,06\text{mV}$ (Tab. 4. 1).

Jako možné vylepšení simulátoru navrhuji implementovat funkci pro simulování zkratu na termočlánek. Další úpravou pro zvýšení celkové úrovně simulátoru by mohla být funkce

pro simulování přerušení vodiče termočlánku. V neposlední řadě bych jako vylepšení umožnil uživateli volbu daného typu termočlánku pro každý kanál zvlášť.

Celkově je zařízení plně funkční a splňuje veškeré požadavky zadavatele práce, jímž je firma Continental Automotive Czech Republic s.r.o.

7 Seznam použité literatury

1. SCHAUER, František a Július KREMPASKÝ. Fyzika I a II. Univerzita Tomáše Bati ve Zlíně a Slovenská technická univerzita v Bratislavě, 2004. Dostupné z: [http://kf-
lin.elf.stuba.sk/KrempaskyFyzika/43.pdf](http://kf.lin.elf.stuba.sk/KrempaskyFyzika/43.pdf)
2. KREIDL, Marcel. Měření teploty: senzory a měřicí obvody. 1. vyd. Praha: BEN - technická literatura, 2005, 239 s. Senzory neelektrických veličin. ISBN 80-730-0145- 4
3. GOLAB, František a František KAMENČÁK. *Termoelektrické jevy a jejich užití*. 1. vyd. Praha: SPN, 1975, 83 s. Pomocné knihy pro žáky. Škola mladých fyziků, sv. 8.
4. Termoelektrický jev. *Know* [online]. [cit. 2017-03-18]. Dostupné z: https://www.know.cf/enciclopedia/CS/Termoelektrick%C3%BD_jev
5. BUREŠ, Jiří. Seebeckův jev. ConVERTER: Převody jednotek, fyzikální tabulky, životopisy fyziků a Nobelova cena. [online]. © 2002 [cit. 2015-01-19]. Dostupné z: <http://www.converter.cz/tabulky/seebeck.htm>
6. NOVÉ TECHNOLOGIE- VÝZKUMNÉ CENTRUM. Termomechanika technologických procesů [online]. Plzeň: Západočeská univerzita v Plzni, © 2000-2014 [cit. 2015-01-19]. Dostupné z: <http://ttp.zcu.cz/cz/uvod>
7. MIKYŠKA, Ladislav. Termoelektrické články. 1. vyd. Praha: Státní nakladatelství technické literatury, 1964, 152 s.
8. ČERMÁK, Patrik. Elektrina přímo z tepla. 3pól [online]. [cit. 2014-11-30]. Dostupné z: <http://3pol.cz/888-elektrina-primo-z-tepla>
9. *Termoelektrické snímače teploty* [online]. [cit. 2017-03-18]. Dostupné z: <http://www.technotrend.cz/stranky/teorie/teorie2.htm>
10. VLACH, Libor. MODERNÍ MĚŘENÍ A REGULACE ENERGETICKÝCH VELIČIN [online]. Brno, květen 2003 [cit. 2015-01-19]. Dostupné z: [http://www.e-
automatizace.cz/ebooks/mmr/v/](http://www.e-automatizace.cz/ebooks/mmr/v/). Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Vedoucí práce Ing. František Vdoleček, CSc.
11. MAREK, Pavel. MM Průmyslové spektrum. MM Průmyslové spektrum [online]. Vestec: MM Průmyslové spektrum, © 2015 [cit. 2015-01-19]. Dostupné z: <http://www.mmspektrum.com/clanek/vyroba-platinovych-termoclanku.html>
12. TIŠNOVSKÝ, Pavel. Externí sériové sběrnice SPI a I²C. Root. 2008. Dostupné z: <https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>
13. 16-Bit Oscillator System: 80 MHz PLL. *Http://microchipdeveloper.com* [online]. 2017 [cit. 2017-04-02]. Dostupné z: <http://microchipdeveloper.com/16bit:osc-pll-80mhzpll>

14. *U1401B Handheld Multi-Function Calibrator/Meter* [online]. 10. Malaysia: © Agilent Technologies, 2014 [cit. 2017-04-14]. Dostupné z: <https://www.valuetronics.com/Manuals/HP-U1401B.pdf>

Seznam příloh

I. Obvodové schéma simulátoru, formát A3