

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

USB TEPLOMĚR

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ HOLUB

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

USB TEPLOMĚŘ

USB THERMOMETER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JIŘÍ HOLUB

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. IVO LATTENBERG, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jiří Holub

ID: 158144

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

USB teploměr

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a vyrobte elektronický USB teploměr, který bude měřit teplotu pomocí čidla v rozsahu teplot alespoň -40 až 120 stupňů Celsia. Zařízení bude napájené z USB portu a bude pracovat i při vypnutí počítače, za předpokladu, že port bude dále napájený. Data budou ukládány do paměti. Vytvořte program pro PC, který bude umožňovat data z USB teploměru vyčíst a prohlížet časové průběhy naměřených hodnot teploty. Ovládací program bude možno také použít ke konfiguraci teploměru.

DOPORUČENÁ LITERATURA:

- [1] BRTNÍK, B., MATOUŠEK, D. Mikroprocesorová technika BEN - technická literatura, Praha 2011, 152 stran, ISBN 978-80-7300-406-4.
- [2] MATOUŠEK, D. Práce s mikrokontroléry Atmel AVR. Nakladatelství BEN - technická literatura, Praha 2006, 376 stran, ISBN 80-7300-209-4.
- [3] VIRIUS, M., C# 2010 Hotová řešení, Computer Press, 2012, 424 s., ISBN 978-80-251-3730-7

Termín zadání: 9.2.2015

Termín odevzdání: 2.6.2015

Vedoucí práce: doc. Ing. Ivo Lattenberg, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá problematikou elektronického měření teploty. Popisuje různé typy teplotních čidel a zpracování z nich získaných dat. Stručně popisuje rozhraní USB a jeho parametry. Dále se zabývá návrhem hardwaru a softwaru teploměru s teplotním čidlem SMT 160, řízený mikrokontrolérem ATmega16 a komunikujícím s PC pomocí rozhraní USB. Navržený teploměr také umožňuje ukládání naměřených dat do zabudované paměti a jejich pozdější načtení. Údaje o teplotě se budou zároveň zobrazovat na LCD. Poslední částí práce je návrh aplikace pro PC, umožňující zobrazení výsledků měření a nastavení teploměru.

KLÍČOVÁ SLOVA

Teplota, teplotní čidla, teploměr, USB rozhraní, USART rozhraní, TWI rozhraní, paměť EEPROM, FTDI, mikrokontrolér ATmega16, SMT 160, LCD, Assembler, C#

ABSTRACT

This bachelor thesis deals with a problem of electrical measurement of temperature. It describes a various types of temperature sensors and processing data obtained from these. It describes in short the USB interface and its parameters. It also contains a design of hardware and software solution of thermometer using temperature sensor SMT 160, managed by microcontroller ATmega16 and communicating with PC via USB interface. Designed thermometer also offers possibility to save measured data to built-in memory and later load them from there. Measured data will be also displayed on LCD. Last part of these thesis is design of application for PC, which will serve for displaying measured data and adjustment of thermometer.

KEYWORDS

Temperature, temperature sensor, thermometer, USB interface, USART interface, TWI interface, EEPROM memory, FTDI, microcontroller ATmega16, SMT 160, LCD, Assembly, C#

HOLUB, Jiří *USB teploměr*. bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 60 s. Vedoucí práce byl doc. Ing. Ivo Lattenberg, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „USB teploměr“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Ivu Lattenbergovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Také bych chtěl poděkovat panu Lukáši Pazderovi, zaměstnanci Ústavu telekomunikací, za výrobu desek plošných spojů pro můj výrobek.

Brno

.....

podpis autora(-ky)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

podpis autora(-ky)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	11
1 Měření teploty	12
1.1 Teplota	12
1.1.1 Definice teploty	12
1.1.2 Historie měření teploty	12
1.2 Teplotní senzory	13
1.2.1 Kovové odporové senzory teploty	13
1.2.2 Polovodičové odporové senzory teploty	15
1.2.3 Monolitické PN senzory teploty	16
1.2.4 Termoelektrické články	17
1.2.5 Senzory s digitálním výstupem	18
2 USB rozhraní	19
2.1 Historie	19
2.2 Kably a konektory	19
2.3 Přenos dat	20
2.4 Rozbočovače	21
2.5 Napájení ze sběrnice USB	21
2.6 Alternativní rozhraní	21
2.6.1 RS-232	21
2.6.2 FireWire	22
2.6.3 Thunderbolt	22
3 Návrh zařízení	23
3.1 Napájení	23
3.2 Teplotní senzor SMT 160-30	24
3.2.1 Měření střídavé	24
3.3 Mikrokontrolér ATmega16	25
3.3.1 Programování mikrokontroléru	27
3.4 Komunikace s počítačem	28
3.4.1 Popis modulu MM232R	28
3.4.2 Sériové rozhraní USART	30
3.4.3 Použití sériového kanálu	30
3.5 Paměť pro data	31
3.5.1 Popis paměti	31
3.5.2 Komunikace s pamětí	32

3.6	LCD displej	33
3.6.1	Popis displeje	34
3.6.2	Komunikace s displejem	35
4	Programové řešení	36
4.1	Program pro mikrokontrolér	36
4.1.1	Měření teploty	37
4.1.2	Zobrazení teploty na displeji	38
4.1.3	Uložení dat do paměti	39
4.1.4	Komunikace s PC	40
4.2	Aplikace pro PC	42
4.2.1	Inicializace rozhraní USB	42
4.2.2	Příjem dat z teploměru	43
4.2.3	Popis uživatelského rozhraní	44
5	Konstrukce	45
	Závěr	46
	Literatura	47
	Seznam symbolů, veličin a zkratk	49
	Seznam příloh	50
A	Schéma hlavní desky	51
B	Deska plošného spoje hlavní desky	52
C	Schéma LCD modulu	53
D	Deska plošného spoje LCD modulu	54
E	Seznam součástek	55
F	Příklad zápisu do paměti	56
G	Fotografie výrobku a screenshoty z aplikace	57
H	Obsah přiloženého CD	60

SEZNAM OBRÁZKŮ

1.1	Teplotní závislost odporu vybraných teplotních snímačů [2]	13
1.2	Teplotní závislost VA charakteristiky PN přechodu diody [1]	16
1.3	Tranzistorová dioda	17
1.4	Termoelektrický jev	17
2.1	USB konektory [7]	20
2.2	Kódování NRZI	21
2.3	Srovnání přenosových rychlostí	22
3.1	Blokové schéma teploměru	23
3.2	Pouzdro mikrokontroléru ATmega16 [13]	26
3.3	Zapojení pro sériové programování [14]	27
3.4	Rozložení vývodů modulu [15]	28
3.5	Asynchronní sériový přenos	30
3.6	Pouzdro paměti [16]	32
3.7	Zápis dat [16]	32
3.8	Čtení dat z aktuální adresy [16]	33
3.9	Čtení dat z nastavené adresy [16]	33
3.10	Seznam příkazů pro ovládání LCD [18]	35
4.1	Zjednodušený vývojový diagram programu pro mikrokontrolér	36
4.2	Registr TWCR pro řízení rozhraní TWI	39
4.3	Screenshot hlavního okna aplikace	44
5.1	Fotografie výrobku	45
A.1	Schéma hlavní desky	51
B.1	Hlavní deska, strana spojů	52
B.2	Hlavní deska, strana součástek	52
C.1	Schéma modulu displeje	53
D.1	Modul displeje, strana spojů	54
D.2	Modul displeje, strana součástek	54
G.1	Screenshot okna s nastavením	57
G.2	Screenshot okna s načtenými daty	57
G.3	Fotografie výrobku, pohled zepředu	58
G.4	Fotografie výrobku, pohled zezadu	58
G.5	Fotografie výrobku, čidlo	59

SEZNAM TABULEK

2.1	Standardy a teoretické přenosové rychlosti USB	19
3.1	Popis vývodů modulu	29
3.2	Vývody displeje	34
4.1	Příkazy od PC	41
4.2	Význam bitů v nastavovacím bytu	42
E.1	Seznam součástek	55

ÚVOD

Zadáním této práce je navrhnout a vyrobit elektronický teploměr, který bude komunikovat s počítačem pomocí sběrnice USB. Naměřená data se budou také zobrazovat na LCD. Když nebude zapnutý obslužný program na počítači nebo bude teploměr napájen z jiného zdroje, např. nabíječky od mobilního telefonu, budou se naměřené hodnoty ukládat do paměti, odkud je bude možné později načíst.

Tento teploměr není navržen pro konkrétní použití, využít bude nejspíše v domácnostech pro měření vnější nebo vnitřní teploty.

Elektronické měření teploty je v dnešní době obsáhlý obor, neboť se využívá ve všech oblastech průmyslu, v lékařství i v každodenním životě obyčejného člověka. Existuje velké množství různých čidel, přičemž každé je vhodné pro jinou aplikaci. Touto problematikou se zabývá první kapitola.

Druhá kapitola pojednává o rozhraní USB, které je nejpoužívanější pro připojování všemožných periférií a přístrojů k počítači. Jsou zde popsány jeho parametry, používané kabely a konektory ale i možné alternativy, o kterých nejspíše málokdo ví, že vůbec existují, i když nabízejí lepší parametry než USB.

Třetí kapitola se zabývá návrhem hardwarové části elektronického teploměru podle výše zmíněného zadání. Jsou zde popsány vlastnosti a funkce použitých součástek jako jsou mikrokontrolér, teplotní čidlo, paměť, displej a převodník USB na sériovou linku. Dále je zde popsáno vzájemné propojení těchto součástek a také jejich ovládání, případně komunikace s nimi.

V následující kapitole je popsán návrh softwaru jak pro mikrokontrolér, v jazyce Assembler, který řídí celý teploměr, tak pro aplikaci na počítači v jazyce C#, sloužící k prezentaci naměřených dat a nastavení teploměru. Jsou zde popsány jednotlivé bloky těchto programů včetně krátkých ukázek kódu, také je zde uveden komunikační protokol mezi teploměrem a počítačem.

Poslední kapitola stručně popisuje konečný vzhled teploměru a jeho konstrukční řešení.

Na konci dokumentu jsou přiložena schémata a návrhy desek plošných spojů z programu Eagle, některé zajímavé ukázky zdrojového kódu a fotografie výrobku v provozu.

1 MĚŘENÍ TEPLoty

1.1 Teplota

Teplota patří mezi nejdůležitější parametry pro zajištění optimálního technologického procesu, zároveň je fyzikální veličinou, se kterou se v běžném životě setkáváme asi nejčastěji a každý ji pocituje na vlastní kůži a také je důležitou veličinou v lékařství, kde je teplota lidského těla používána k diagnostice onemocnění. Teplota je také základní fyzikální veličinou soustavy SI. Základní jednotkou teploty je Kelvin (K).

1.1.1 Definice teploty

Teplota je stavová veličina určující stav termodynamické rovnováhy tj. stav, kdy v izolované soustavě těles od okolního prostředí neprobíhají žádné makroskopické změny a všechny fyzikální veličiny, jimiž je stav soustavy popsán nezávisí na čase [1].

Jedná se tedy o skalární veličinu popisující stav ustálených makroskopických systémů a v podstatě vyjadřující míru kinetické energie pohybu atomů a molekul v dané látce.

Teplota je jednou z mála fyzikálních veličin, která se nedá měřit přímo, ale pouze prostřednictvím jiných veličin. Zároveň málokteré fyzikální jevy nebo vlastnosti látek jsou na teplotě nezávislé.

1.1.2 Historie měření teploty

Historicky první popsany teploměr byl založen na roztažnosti vzduchu, byl vynalezen italským fyzikem Galileo Galileiem v roce 1592, byl však málo přesný a závislý na atmosférickém tlaku. Další významný milník ve vývoji teploměru nastal v roce 1724 vynalezením rtuťového teploměru německým fyzikem Fahrenhaitem využívajícím první teplotní stupnici ($^{\circ}\text{F}$).

Roku 1730 navrhuje svou stupnici také francouzský přírodovědec Réaumur ($^{\circ}\text{R}$) a dále pak v roce 1742 přichází se svou stupnicí také švédský astronom Celsius ($^{\circ}\text{C}$). Tato stupnice se stává nejpoužívanější pro běžné měření teploty. A konečně v roce 1848 britský fyzik lord Kelvin zavedl termodynamickou stupnici (K).

Prvním použitím teploměru v praxi byl lékařský teploměr (1866). V současnosti se bez měření teploty neobejdeme téměř v žádném oboru lidské činnosti, např. v energetice, dopravě, průmyslové produkci elektrospotřebičů, výpočetní techniky atd.

Pro měření teploty se nejčastěji využívá těchto fyzikálních jevů:

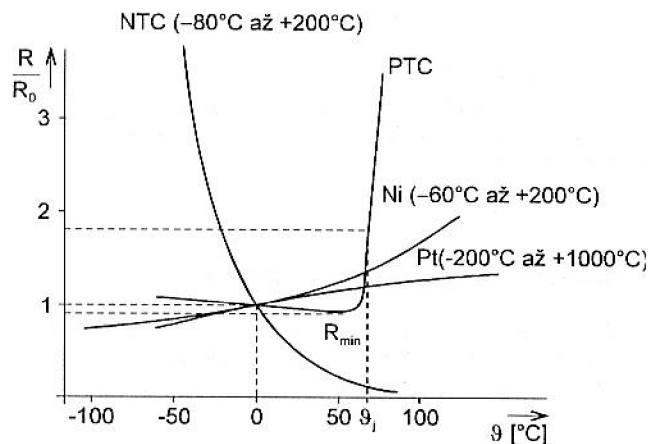
- změna objemu různých kapalin a plynů (rtuť, líh),
- délkové roztažnosti různých kovů spojených v tzv. bimetalový pásek, kdy se při změně teploty tento pásek ohýbá,
- změny tlaku plynů při konstantním objemu,
- změna elektrických vlastností materiálů, viz kapitola 1.2.

1.2 Teplotní senzory

Teplotní senzory jsou zařízení, která převádějí neelektrickou veličinu (teplotu) na elektrickou veličinu (elektrický odpor, napětí, střída). Můžeme je dělit podle různých kritérií:

- podle způsobu měření: kontaktní, bezkontaktní,
- podle fyzikálního principu: odporové, termoelektrické, polovodičové s PN přechodem, dilatační, optické, radiační, chemické,
- podle transformace signálu: aktivní (chovají se jako zdroj elektrické energie), pasivní (potřebují napájení),
- podle typu výstupního signálu: analogové, digitální [1].

V následující části této kapitoly budou podrobněji popsány některé kontaktní teplotní senzory na základě fyzikálního principu.



Obr. 1.1: Teplotní závislost odporu vybraných teplotních snímačů [2]

1.2.1 Kovové odporové senzory teploty

Princip těchto senzorů je postaven na teplotní závislosti odporu kovu. Základní materiálovou konstantou je teplotní součinitel odporu α . Pro rozsah teplot 0 – 100 °C

lze použít s určitou nejistotou lineární vztah

$$R_t = R_0(1 + \alpha t), \quad (1.1)$$

kde R_0 je odpor čidla při teplotě 0 °C

Pro větší rozsahy platí nelineární rovnice v závislosti na použitém materiálu. Nejpoužívanější materiály pro odporová čidla jsou platina, nikl a měď [3].

Měřicí obvody pro vyhodnocování odporových snímačů musí splňovat určitá kritéria:

- vliv měřicího proudu procházejícího čidlem musí být minimální,
- vliv odporu vedení musí být minimální,
- ve většině aplikací se musí závislost odporu na teplotě linearizovat (analogově nebo číslicově).

Platinové odporové senzory

Platina se vyznačuje chemickou netečností, časovou stálostí a vysokou teplotou tání. Pro metrologické účely jsou nároky na čistotu velmi vysoké (až 99,999 %). Dle doporučení IEC-751 se platinové odpory pro průmyslové použití dělí do dvou tříd. Třída A pro rozsah -200 až 650 °C a třída B pro rozsah -200 až 850 °C . Dále se ještě vyrábějí vysokoteplotní snímače až do 1100 °C [1]. Pro rozsah teplot od -200 do 0 °C je závislost odporu dána vztahem

$$R_t = R_0[1 + At + Bt^2 + Ct^3(t - 100)] \quad (1.2)$$

a pro teploty od 0 do 850 °C vztahem

$$R_t = R_0(1 + At + Bt^2), \quad (1.3)$$

kde

$$\begin{aligned} R_0 &= 100\ \Omega \\ A &= 3,90802 \cdot 10^{-3}\ \text{K}^{-1}, \\ B &= -5,80195 \cdot 10^{-7}\ \text{K}^{-2}, \\ C &= -4,27350 \cdot 10^{-12}\ \text{K}^{-4}\ \text{pro } t < 0\ \text{°C}, \\ C &= 0\ \text{pro } t > 0\ \text{°C}. \end{aligned}$$

Kromě základní hodnoty odporu $R_0 = 100\ \Omega$ se vyrábějí i v hodnotách 50, 200, 500, 1000 a 2000 Ω [1].

Niklové odporové senzory

Niklové senzory mají oproti platinovým výhodu v tom, že jsou citlivější, mají rychlejší odezvu a menší rozměry. Nevýhodou však je menší teplotní rozsah a větší nelinearita. Základní hodnota odporu je $R_0 = 100\ \Omega$, vyrábí se však i s hodnotami 200,

500, 1000 a 2000 Ω . Teplotní závislost odporu lze vyjádřit polynomem

$$R_t = R_0(1 + At + Bt^2 + Ct^4 + Dt^6), \quad (1.4)$$

kde

$$A = 5,485 \cdot 10^{-3} \text{ K}^{-1},$$

$$B = 6,65 \cdot 10^{-6} \text{ K}^{-2},$$

$$C = 2,805 \cdot 10^{-11} \text{ K}^{-4},$$

$$D = 2,0 \cdot 10^{-17} \text{ K}^{-6} [1].$$

Měděné odporové senzory

Tyto senzory se běžně nevyrábějí kvůli malé rezistivitě a snadné oxidaci. Výhodou však je např. možnost přímého měření teploty vinutí elektrických strojů prostřednictvím měření odporu vinutí [1].

1.2.2 Polovodičové odporové senzory teploty

Tyto senzory využívají také závislost odporu na teplotě, hlavní je však závislost koncentrace nosičů n náboje dle vztahu

$$n = e^{-\frac{\Delta E}{2kT}}, \quad (1.5)$$

kde ΔE je šíře mezery mezi energetickými hladinami, k je Boltzmanova konstanta.

Polovodičové odporové senzory dělíme na:

- termistory
 - negastory
 - pozistory
- monokrystalické odporové senzory

Termistory

Termistor je teplotně závislý odpor vyrobený z polovodičových materiálů. Jeho výhody jsou velká teplotní citlivost a malé rozměry, nevýhodou pak horší stabilita, menší teplotní rozsah a velká nelinearita. Tato nelinearita se však dá dostatečně přesně linearizovat zapojením paralelního nebo sériového odporu. V závislosti na materiálu má termistor teplotní součinitel odporu buď záporný (negastor, NTC – Negative Temperature Coefficient) nebo kladný (pozistor, PTC – Positive Temperature Coefficient). Jejich teplotní závislosti jsou na obrázku 1.1.

Negastory jsou použitelné pro rozsahy teplot od -50 do 150 $^{\circ}\text{C}$, případně pro extrémní podmínky od $4,2$ do 1000 $^{\circ}\text{C}$.

Teplotní závislost odporu negastoru můžeme vyjádřit vztahem

$$R_T = A + e^{\frac{B}{T}}, \quad (1.6)$$

kde

A je konstanta závislá na geometrickém tvaru a materiálu termistoru,

B je teplotní konstanta daná materiálem termistoru,

T je termodynamická teplota (K) [1].

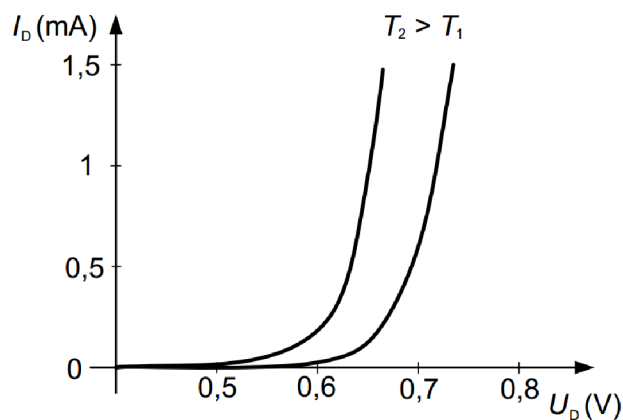
Negastory se nejčastěji používají v méně náročných aplikacích, pro měření rychlých změn teploty v menším rozsahu teplot.

Pozistory jsou termistory s kladným teplotním součinitelem. Jejich teplotní závislost nejprve mírně klesá, poté prudce stoupá, následně opět mírně klesá (v této oblasti se obvykle nepoužívá). Díky prudkému nárůstu odporu při určité teplotě se nejčastěji využívají jako dvoustavové senzory, např. pro signalizaci překročení určité teploty, kde je zlomová teplota dána chemickým složením pozistoru [4].

Monokrystalické Si senzory

Jsou založeny na změně pohyblivosti nosičů náboje v polovodičích, s rostoucí teplotou pohyb ustává a tím se zvyšuje odpor. Výhodou těchto senzorů je linearita a stabilita lepší než u termistorů, ale horší než u platinových senzorů, nevýhodou menší teplotní rozsah a menší citlivost než u termistorů. Nejčastěji se používají v rozsahu -50 až 150 °C. V méně náročných aplikacích mohou nahradit kovové odporové senzory, je však nutno počítat s linearizací. Typickými představiteli těchto senzorů jsou čidla řad KY a KTY [4].

1.2.3 Monolitické PN senzory teploty

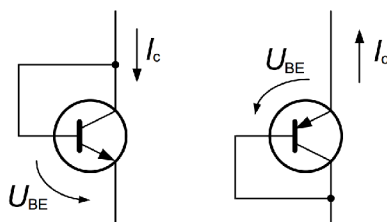


Obr. 1.2: Teplotní závislost VA charakteristiky PN přechodu diody [1]

Jejich funkce je založena na teplotní závislosti PN přechodu v propustném směru. Obvyklý teplotní rozsah těchto senzorů je od -55 do 150 °C. Jako čidel se využívá diod a tranzistorů.

Teplotní součinitel je záporný, obvykle v rozmezí $-2,5$ až -2 mV/K, s rostoucí teplotou se tedy snižuje napětí na diodě, viz obrázek 1.2.

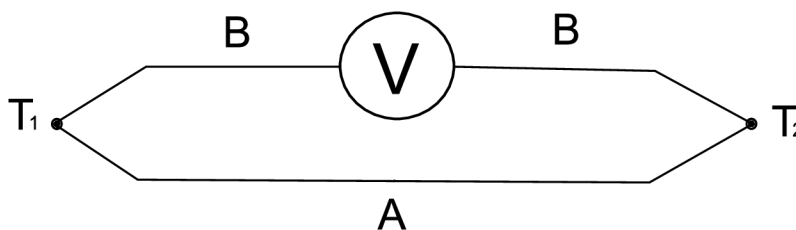
Tranzistor se používá v zapojení tzv. tranzistorové diody dle obrázku 1.3. Využívá se teplotní závislosti napětí přechodu báze-emitorek v propustném směru [1].



Obr. 1.3: Tranzistorová dioda

1.2.4 Termoelektrické články

Termoelektrické články využívají principu Seebeckova jevu, tedy na převodu tepelné energie na elektrickou. Termoelektrický článek se skládá ze dvou vodičů z různých materiálů, které jsou na obou koncích spojeny. Tímto obvodem protéká proud, pokud mají oba spoje rozdílnou teplotu. Když tento obvod na kterémkoli místě rozpojíme, na vzniklých svorkách naměříme termoelektrické napětí.



Obr. 1.4: Termoelektrický jev

Jeden ze spojů je měřicí, druhý je srovnávací (jeho teplota se používá pro stanovení základních hodnot napětí). Jednotlivé páry materiálů jsou normované, vybírané na základě parametrů jako linearita, odolnost proti korozi, chemickým vlivům a stabilita charakteristiky v čase.

Jejich výhodou je velký teplotní rozsah, nevýhodou je malá citlivost (malé výstupní napětí) a složitější připojení k měřicímu obvodu tak, aby neovlivňoval výsledky měření. Používají se nejčastěji pro měření velkých změn teplot [1].

1.2.5 Senzory s digitálním výstupem

Tyto senzory jsou v podstatě také analogové, využívají některého z předchozích fyzikálních principů, rozdíl je pouze v tom, že v jednom pouzdře obsahují jak čidlo, tak linearizační obvody a A/D převodník, takže výstupem je digitální signál ve kterém je informace přenášena např. pomocí pulsně šířkové modulace nebo pomocí standardních sběrnic (I²C, 1-Wire) jako n-bitová hodnota.

Příklady běžně dostupných senzorů s digitálním výstupem:

- SMT160 – převodník teplota – šířka impulsu, výstupní signál se dá snadno zpracovat mikroprocesorem nebo převést na analogový signál, teplotní rozsah -45 °C až 130 °C ,
- DS18B20 – využívá sběrnici 1-Wire, výstupem je 9 až 12-ti bitové digitální slovo, teplotní rozsah -55 °C až 125 °C .

Výhodou digitálních senzorů teploty je nižší cena a snadnější zpracování výsledků, nevýhody jsou menší přesnost, teplotní rozsah, životnost a stabilita.

2 USB ROZHRAŇÍ

USB (Universal Serial Bus) je sériová sběrnice, data se tedy odesílají bit po bitu. Na rozdíl od RS232 se data přenášejí v obou směrech po stejných linkách a to diferenčně (vodiče D+ a D−). Rozdíl napětí mezi datovými vodiči je 3,3 V. Jedná se o sběrnici typu master – slave, master je vždy jen jeden (počítač), slave zařízení může být více, každé dostane přiřazenou svou adresu, je tedy možné se na ně cíleně obracet [5].

Rozhraní také podporuje technologii Plug&Play, systém tedy rozpozná nově připojená zařízení a automaticky nainstaluje ovladače.

2.1 Historie

USB vzniklo v roce 1995 jako náhrada sériového portu RS-232 pro připojení externích zařízení. Velkého rozšíření se dočkalo v roce 1998 uvedením počítače iMac od Applu na trh. Tento počítač byl totiž vybaven jenom porty USB a díky jeho velké oblibě se začalo vyrábět více periferií s tímto rozhraním [6].

V dnešní době se používá pro připojení většiny periferií k počítači (klávesnice, myš, tiskárna, externí disky), ale i k propojení dalších přístrojů s počítačem (teploměry, zabezpečovací ústředny).

2.2 Kabely a konektory

Ve starších verzích USB (1.1 a 2.0) využívalo čtyři vodiče (U_{CC} , GND, Data+ a Data−), přičemž datové vodiče jsou kroucené, napájecí nejsou. Ve verzi 3.0 využívá vodičů 9, přičemž zachovává zpětnou kompatibilitu se staršími konektory. Typy USB konektorů jsou na obrázku 2.1.

Tab. 2.1: Standardy a teoretické přenosové rychlosti USB

Standard	Přenosová rychlost	Režim
USB 1.1	1,5 Mb/s	Low-Speed
USB 1.1	12 Mb/s	Full-Speed
USB 2.0	480 Mb/s	Hi-Speed
USB 3.0	5 Gb/s	Super-Speed
USB 3.1	10 Gb/s	Super-Speed+



Obr. 2.1: USB konektory [7]

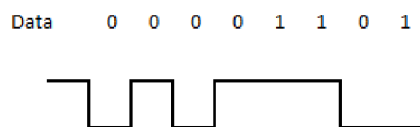
V blízké době bude na trh uvedena nová verze USB 3.1, která nabízí rychlost až 10 Gb/s a nový konektor, který sice nebude kompatibilní se staršími verzemi, díky oboustranné konstrukci však nebude záležet na tom, jak jej do zásuvky zasunete.

2.3 Přenos dat

Jak již bylo napsáno výše, veškerá komunikace vychází od počítače. To může data vysílat nebo je požadovat od připojených zařízení, ta však nemohou odesílat sama od sebe. Pokud potřebuje počítač komunikovat s více než jedním zařízením, o rozdělení paketů se stará rozbočovač (hub).

Data se přenášejí v paketech o délce 8 – 64 bytů. Jelikož se v přenosu nepoužívá hodinový signál, data jsou kódována metodou NRZI (Non Return To Zero). Při logické 0 v datech dochází ke změně úrovně, při logické 1 zůstává stávající úroveň. Kódování a dekódování je pouze hardwarovou záležitostí, programátor ovladačů se o to již nemusí starat, odesílá a získává proud bitů. Za účelem synchronizace obsahuje každý paket tzv. synchronizační byte (00000001) díky kterému se přijímač jednoduše zasynchronizuje [8].

Přijímač i vysílač jsou vždy v jedné součástce. Obsahují paměť typu FIFO (First In, First Out), tzv. buffer, kam se vysílaná (přijímaná) data ukládají před dalším zpracováním.



Obr. 2.2: Kódování NRZI

2.4 Rozbočovače

Pro připojení více než jednoho zařízení se používá rozbočovačů. Ty mohou být buď uvnitř počítače (tzv. kořenový hub, zajišťuje možnost realizace více portů) nebo může být rozbočovač externí. Ten je připojen k USB portu počítače a umožňuje připojení dalších zařízení a tak rozšiřuje počet připojitelných zařízení. Externí rozbočovače se dále dělí na pasivní a aktivní. Ty mají navíc napájení a umožňují tak dodávat do každého připojeného zařízení maximální proud [8].

2.5 Napájení ze sběrnice USB

Rozhraní USB poskytuje stejnosměrné napětí 5 V a do zařízení může dodávat proud 100 mA. Po enumeraci (ohlášení) zařízení do operačního systému může požádat o větší proud, maximálně však 500 mA. Systém poté rozhodne, je-li k dispozici dostatek rezerv a může povolit zvýšení proudu, v opačném případě může žádost zamítnout [5].

2.6 Alternativní rozhraní

Rozhraní USB má několik konkurentů, avšak nabízí vyvážený poměr rychlosti, ceny a náročnosti pro vývojáře a navíc je masově rozšířený, takže bude i nadále nejpoužívanějším rozhraním. Konkurenční rozhraní se využívají spíše v profesionální oblasti nebo u úzkého segmentu trhu.

2.6.1 RS-232

Dříve se hojně používalo sériové rozhraní RS-232 pro propojení počítačů a různé další elektroniky. V dnešní době je již téměř všude nahrazeno USB. RS-232 je jednodušší na implementaci, ale je pomalé a nenabízí možnost napájení zařízení.

2.6.2 FireWire

FireWire je sériová sběrnice pro připojení periférií k počítači. Je vyvinuta společností Apple. Existuje ve dvou verzích FireWire 400 a 800 (přenosové rychlosti 400 a 800 Mb/s). Nabízí tedy vyšší přenosovou rychlost než USB 2.0. Nabízí také možnost napájení připojeného zařízení napětím 30 V při proudu až 1,5 A. Nevýhodou oproti USB jsou dražší kabely a několik nekompatibilních konektorů.

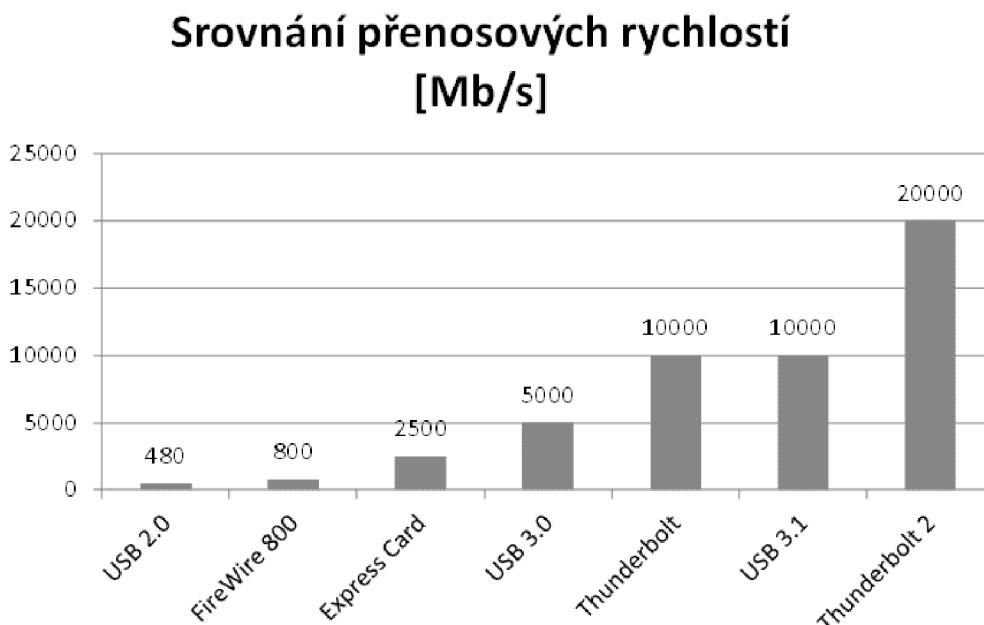
FireWire je málo rozšířen a v dohledné době tomu nebude jinak. Používá se spíše v profesionální sféře pro připojení videokamer, externích disků nebo čteček paměťových karet [9].

2.6.3 Thunderbolt

Je rozhraní vyvinuté společností Intel, využívané převážně v produktech společnosti Apple. Současná verze Thunderbolt 2 dosahuje přenosové rychlosti až 20 Gb/s, třetí generace, která má být uvedena na trh v roce 2015, slibuje rychlost až 40 Gb/s.

Thunderbolt umožňuje kromě připojení periférií jako externí disky také připojení monitorů. Umožňuje také napájení připojených zařízení až 10 W.

Výhodou je tedy vysoká rychlost a univerzálnost, nevýhodou vysoká cena (2m kabel stojí 999 Kč). Má však velký potenciál, je totiž volně dostupné bez licencování, což ve své době zabránilo masovějšímu rozšíření jinak vynikajícího FireWire [10].



Obr. 2.3: Srovnání přenosových rychlostí

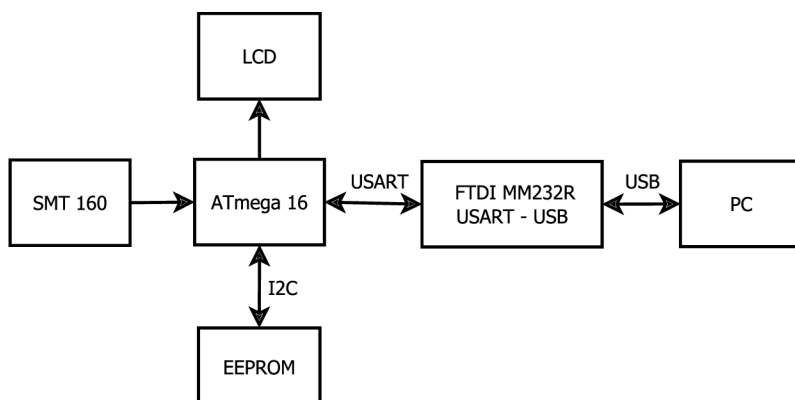
3 NÁVRH ZAŘÍZENÍ

Řídicím prvkem celého teploměru je mikrokontrolér ATmega 16. Jako teplotní čidlo byl zvolen integrovaný teplotní senzor SMT 160-30 s výstupem ve formě pulsně šířkově modulovaného signálu, kdy střída signálu závisí na teplotě podle rovnice 3.1.

Mikrokontrolér bude ze střídy signálu počítat teplotu, kterou bude zobrazovat na LCD displeji WINSTAR WH1602A-TTI-ET a následně odesílat přes USB rozhraní do PC, kde se bude dále zpracovávat. Jelikož mikrokontrolér neumí komunikovat přímo přes USB rozhraní, je zde použit převodník USART – USB MM232R od firmy FTDI.

V případě vypnutí počítače nebo obslužného programu, pokud bude port USB stále napájen nebo bude využito jiného napájení, budou se data ukládat do paměti 24LC512-I/P typu EEPROM s kapacitou 64 kB, která je k mikrokontroléru připojena sběrnici I²C.

Blokové schéma teploměru je na obrázku 3.1, podrobnější informace o použitých součástkách naleznete v další části této kapitoly.



Obr. 3.1: Blokové schéma teploměru

3.1 Napájení

Teploměr bude napájen ze sběrnice USB, která poskytuje napětí 5 V a proud až 500 mA, což je pro toto zařízení více než dostatečné. Při resetování PC může docházet ke krátkodobému výpadku napájení USB portů, proto je na vstupu napájení teploměru umístěn elektrolytický kondenzátor s kapacitou 1 mF/16 V, který by měl udržet zařízení v chodu po dobu tohoto výpadku.

3.2 Teplotní senzor SMT 160-30

Jak již bylo zmíněno výše, jedná se o integrovaný teplotní senzor s číslicovým výstupem od firmy SMARTEC. Výstupem je obdélníkový signál, jehož střída se mění lineárně s teplotou. Střída (poměr doby trvání logické 1 k době periody) je dána rovnicí

$$D.C. = 0,320 + 0,00470 \cdot t \quad (3.1)$$

kde $D.C.$ (duty cycle) je střída signálu a t je teplota.

Senzor se vyrábí v pouzdrech TO18, TO92 a TO220. Byl zvolen senzor v pouzdře TO92, který je sice méně přesný, ale jeho cena je méně než poloviční oproti přesnější variantě TO18.

Základním prvkem uvnitř senzoru SMT 160-30 je křemíkový polovodičový přechod PN polarizovaný v propustném směru. Více o tomto způsobu měření teploty najdete v kapitole 1.2.3.

Dále jsou uvnitř senzoru elektronické obvody pro převod signálu na pulsně šířkovou modulaci.

Toto čidlo bylo zvoleno kvůli dostatečné přesnosti pro tuto aplikaci, snadnosti zpracování výstupních dat, nízké ceně a nezávislosti na odporu přívodního kabelu jako odporová čidla, díky čemuž je možné čidlo připojit až na 20m kabel.

Základní parametry

- teplotní rozsah -45 až 130 °C,
- absolutní přesnost $\pm 1,2$ °C,
- nelinearita nižší než $0,4$ °C,
- napájecí napětí $4,75 - 7$ V, napájecí proud maximálně 200 μ A,
- má tři vývody – U_{CC} , GND a číslicový výstup, který je možno připojit přímo na port mikrokontroléru,
- výstupní číslicový signál jednoduše zpracovatelný analogově (po průchodu integračním článkem),
- kalibrováno při výrobě,
- výstupní signál kompatibilní s úrovněmi CMOS i TTL,
- připojení kabelem o délce až 20 m,
- nízká spotřeba (méně než 1 mW) [11].

3.2.1 Měření střídy

Střídu můžeme měřit přímo mikrokontrolérem a to konkrétně vzorkováním obdélníkového výstupního signálu. Čidlo je připojeno na PINB4, měření tedy bude spočívat v kontrole, zda je tento vstup v log. 1 nebo 0. Zde je důležitá vzorkovací frekvence,

obecně platí, že čím vyšší frekvence, tím větší přesnost měření. Dle údajů výrobce je frekvence výstupního signálu v rozmezí 1 – 4 kHz, pracovní frekvence mikrokontroléru je tedy několikanásobně vyšší, takže pro přesné měření bude dostačující. Je zde však problém s tím, že se neměří pulsy od začátku, takže na začátku a na konci měření může zůstat nezměřená část pulsu. Tato chyba se však dá eliminovat větším počtem vzorků v jednom měření a průměrováním hodnot z několika měření. Více o měření teploty pomocí tohoto čidla se dočtete v kapitole 4.1.1 popisující program pro měření.

3.3 Mikrokontrolér ATmega16

Jedná se o 8bitový mikrokontrolér od společnosti Atmel. Využívá architekturu RISC (Reduced Instruction Set Computer) – procesor s redukovanou instrukční sadou. Jeho instrukční sada tedy obsahuje omezené množství instrukcí (konkrétně 131), které jsou však dobře optimalizované a tím zajišťují vysoký výkon.

Tento mikrokontrolér byl zvolen kvůli tomu, že disponuje rozhraními USART a I²C, zde nazývané TWI, které jsou potřeba v tomto projektu. Další důležitým prvkem je zabudovaný 16bitový časovač pro počítání intervalu ukládání dat do paměti. V neposlední řadě je výhodou počet vstupně/výstupních portů, kdy jich po zapojení všech důležitých prvků zbývá dostatek pro případné budoucí rozšíření.

Mezi další klíčové vlastnosti patří:

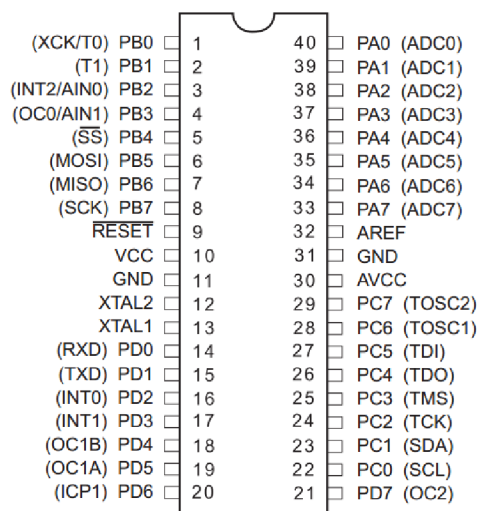
- 32 x 8bitový pracovní registr, připojený přímo k aritmeticko-logické jednotce, což zrychluje práci s těmito registry,
- 16kB paměť Flash pro uložení programu,
- 512B paměť EEPROM pro data,
- 1kB paměť SRAM pro data,
- 2 x 8bitový a 1 x 16bitový čítač/časovač,
- 10bitový A/D převodník,
- rozhraní pro sběrnici I²C, zde nazývanou Two-Wire,
- sériové rozhraní USART,
- sériové rozhraní SPI,
- 32 programovatelných vstup/výstupních pinů rozdělených do čtyř 8bitových portů (PORT A – PORT D),
- napájecí napětí 4,5 – 5,5 V.

Tento mikrokontrolér se vyrábí v provedení DIP i SMD. Bylo zvoleno provedení DIP z důvodu snadnějšího pájení a snadné výměny při osazení do patice.

ATmega16 má 40 vývodů, jejich rozložení a označení funkce je na obrázku 3.2. Většina vývodů má více funkcí – vstup/výstupní porty (PA, PB, PC, PD)

a např. vstup vnějšího přerušení (INT), výstup čítače (OC), vstupy a výstupy pro komunikaci přes sériové rozhraní (RXD, TXD), sběrnici I²C (SDA, SCL), sběrnici SPI (MISO, MOSI, \overline{SS} , SCK), kanály A/D převodníku (ADC). Dalším důležitým vývodem je \overline{RESET} , který po přivedení logické 0 resetuje mikrokontrolér [13].

V tomto zařízení je využit jeden port pro připojení LCD displeje (PORTA), SPI rozhraní pro programování (PORTB 5,6,7), I²C rozhraní pro připojení paměti (PORTC 0,1) a sériové rozhraní pro komunikaci s PC (PORTD 0,1). Ostatní vývody jsou vyvedeny na svorkovnici, kam bude připojeno teplotní čidlo, případně další periferie. Na svorkovnici je vyvedeno i napájecí napětí.



Obr. 3.2: Pouzdro mikrokontroléru ATmega16 [13]

Mikrokontrolér potřebuje pro svou činnost zdroj hodinových impulsů, tzv. časování. Frekvence těchto impulsů udává pracovní frekvenci mikrokontroléru, což je počet provedených instrukcí za sekundu. Jako zdroj časování může být použit zabudovaný RC oscilátor s frekvencí 8 MHz, který však není moc přesný, nebo externí oscilátor v podobě krystalu připojeného mezi vývody XTAL1 a XTAL2. Tyto vývody se zároveň připojí na zem přes keramické kondenzátory s kapacitou 22 pF dle doporučení výrobce. Použití externího oscilátoru nám umožňuje využívat přesnější frekvenci až do 16 MHz.

Volba oscilátoru

V tomto případě tedy určuje pracovní frekvence mikrokontroléru vzorkovací frekvenci pro měření střídavého napětí na výstupu teplotního čidla. Pro vzorkovací frekvenci platí, že čím je vyšší, tím je měření přesnější.

Pracovní frekvence je také důležitá pro nastavení přenosové rychlosti při sériové komunikaci s převodníkem USART – USB. Použitý převodník podporuje standardní

přenosové rychlosti od 300 Bd do 3 MBd. Nastavení přenosové rychlosti v mikrokontroléru probíhá nastavením registru UBRR dle vzorce

$$UBRR = \frac{f_{osc}}{16 \cdot PR} - 1, \quad (3.2)$$

kde f_{osc} je frekvence oscilátoru a PR je přenosová rychlost.

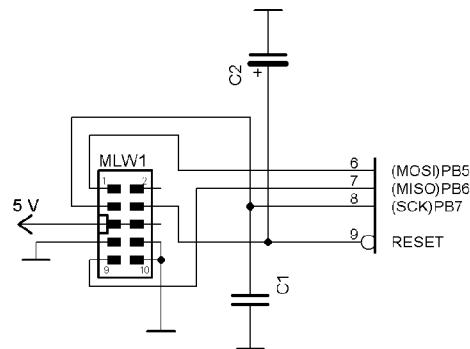
Po dosažení standardní přenosové rychlosti, např. 9600 Bd a hodnot sériově vyráběných krystalů bychom měli dostat celé číslo aby byla synchronizace s převodníkem co nejpřesnější.

Těmto požadavkům odpovídají tzv. baud friendly krystaly např. s frekvencí 11,0592 MHz nebo 14,7456 MHz které umožňují spolehlivou rychlost ve standardních hodnotách až do 230400 Bd. Zde je použit krystal 11,0592 MHz.

3.3.1 Programování mikrokontroléru

Mikrokontroléry AVR je možno programovat dvěma způsoby. Buď paralelně, kdy je nutné mikrokontrolér vložit do programátoru a po naprogramování opět vyjmout a zapojit do zařízení. Tento způsob nabízí více možností nastavení [12].

Další možností je tzv. sériový download, který je použit v této práci. Umožňuje programování mikrokontroléru přímo ve vyvíjené aplikaci, což výrazně urychlí vývoj. K této metodě programování se využívá rozhraní SPI zapojené dle 3.3.



Obr. 3.3: Zapojení pro sériové programování [14]

Jako programátor využívám zařízení ProgAVR, který jsem zakoupil na střední škole. Tento programátor podporuje jak sériové tak paralelní programování a k počítači je připojen USB rozhraním. K tomuto programátoru bylo vyvinuto i vývojové prostředí WinAVR, které umožňuje napsaný program v Assembleru odeslat přímo do mikrokontroléru. Více informací o programátoru a vývojovém prostředí naleznete na [14].

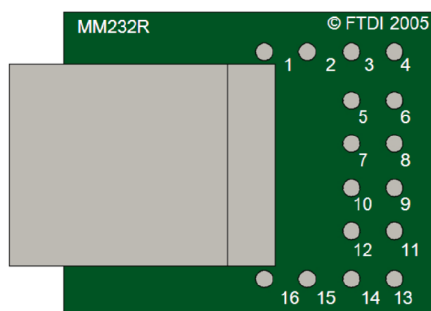
3.4 Komunikace s počítačem

Dle zadání má teploměr odesílat naměřená data do počítače přes rozhraní USB. Přímá implementace komunikace přes USB v mikrokontroléru by byla velmi obtížná ne-li nemožná a proto je výhodné použít převodník USB na sériovou linku.

Existuje velké množství integrovaných obvodů, např. FT232RL nebo CP2102, které toto propojení umožňují. Já jsem se rozhodl použít hotový vývojový modul MM232R od firmy FTDI, využívající obvod FT232RQ. Toto řešení je sice o něco dražší než výroba vlastního převodníku s některým výše zmíněným obvodem, na druhou stranu je však zaručena funkčnost a odpadá problém s pájením těchto obvodů, které jsou vyráběny pouze v provedení SMD s 28 vývody.

3.4.1 Popis modulu MM232R

Na jedné straně se nachází klasický konektor USB 2.0 B pro připojení k počítači běžným kabelem USB 2.0 A-B. Na straně druhé kolíková lišta uspořádaná dle obrázku 3.4, sloužící pro připojení do DPS.



Obr. 3.4: Rozložení vývodů modulu [15]

Modul může být napájen jak ze sběrnice USB, tak z externího zdroje v rozmezí 3,3 – 5,25 V. V případě napájení ze sběrnice se spojí vývody USBPWR a VCC50, externí napájení se přivede na vývod VCC.

Pro samotnou sériovou komunikaci podporuje modul běžné napěťové úrovně 3,3 V a 5 V. Pro volbu úrovně slouží vývod VCCIO, kdy pro úroveň 3,3 V se spojí s vývodem VCC30, pro 5 V se spojí s VCC. Více o funkci a nastavení vývodů naleznete v tabulce 3.1.

Modul dále obsahuje interní EEPROM paměť s kapacitou 1024 bitů, ve které jsou uloženy informace o zařízení a další nastavení modulu. Paměť obsahuje tzv. Vendor ID, tedy identifikátor výrobce, který musí mít každé USB zařízení na trhu. Dále obsahuje název výrobce a popis zařízení. Z nastavení uveďme např. nastavení frekvence hodinového signálu na vývodu CLK12 (6, 12, 24 nebo 48 MHz), invertování úrovní

na vývodech RXD, TXD, RTS#, CTS# a nebo nastavení funkce vývodů PWREN a SLEEP [15].

Tab. 3.1: Popis vývodů modulu

Číslo vývodu	Název vývodu	Popis
1	SHIELD	Stínění kabelu
2	VCC	Vstup externího napájení, při napájení ze sběrnice je na tomto pinu 5 V
3	VCCIO	Určuje napěťové úrovně sériové komunikace, pro 3,3 V spojit s VCC30, pro 5V spojit s VCC
4	VCC30	Výstup napětí 3,3 V
5	TXD	Vysílání dat
6	RTS#	Request to send, pro řízení komunikace
7	RXD	Příjem dat
8	CTS#	Clear to send, pro řízení komunikace
9	CLK12	Výstup hodinového signálu
10	SLEEP#	Indikuje, zda je USB v režimu suspend
11	RESET#	Externí resetování modulu
12	PWREN#	Výstup pro řízení napájení přídavné logiky
13	GND	Zem
14	VCC50	Napájení modulu
15	USBPWR	Napájecí výstup ze sběrnice USB
16	SHIELD	Stínění kabelu

Ovladače

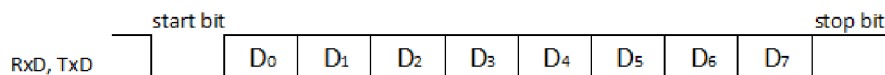
Ze stránek výrobce je možné stáhnout ovladače k tomuto modulu buď ve formě virtuálního COM portu (VCP) nebo jako přímé USB ovladače (D2XX). První způsob se programově obsluhuje jako sériový port, druhý způsob zase nabízí větší možnosti nastavení, připojení více stejných zařízení a přistupování ke každému pomocí přiděleného názvu, ať už je připojený na kterýkoliv port. K této variantě ovladačů je také dostupná knihovna s funkcemi potřebnými pro ovládání modulu. Oba druhy ovladačů jsou dostupné pro všechny běžně používané operační systémy jako Windows, Linux, Mac OS i Android.

3.4.2 Sériové rozhraní USART

Sériový přenos můžeme rozdělit na dva typy:

- synchronní – pro přenos jedním směrem se využívají dva vodiče (DATA a CLK), pro každý přenesený bit proběhne na hodinovém výstupu jeden impuls,
- asynchronní – pro každý směr vysílání se využívá jeden vodič (Tx vysílání, Rx příjem). Přenos začíná START bitem (logická 0), poté následují data (8 nebo 9 bitů) a STOP bit (logická 1).

Mikrokontrolér ATmega16 a převodník USB – USART používají asynchronní přenos. Na pouzdru mikrokontroléru se jedná o vývody PD0 (RXD) a PD1 (TXD), které se propojí s vývody převodníku RXD a TXD do kříže tak, aby vyslaná data z jednoho přišla na vstup druhého. Vývody CTS# a RTS# na převodníku, které jsou určeny pro řízení komunikace, nejsou pro tento případ potřeba a zůstanou nezapojeny. Více o ovládání rozhraní USART se dočtete v kapitole 4.1.4.



Obr. 3.5: Asynchronní sériový přenos

3.4.3 Použití sériového kanálu

Řízení sériového kanálu je v mikrokontroléru prováděno pomocí registrů. Registry UCSRA, UCSRB a UCSRC slouží pro nastavení parametrů přenosu (počet datových bitů, počet STOP bitů, parita) a povolení operací přerušování na rozhraní. Registr UDR slouží pro příjem a vysílání dat. A registr UBRR slouží pro nastavení přenosové rychlosti.

Vysílání

Vysílání začíná zápisem dat do registru UDR, odkud jsou data přenesena do vysílacího posuvného registru. Je-li tento registr prázdný, data jsou okamžitě odeslána a nastaví se příznak UDRE (datový registr prázdný), poté je možné vysílat další byte. Není-li poté do UDR zapsán další byte, nastaví se příznak TXC (přenos kompletní).

Příjem

Je-li povolen příjem, vzorkuje se signál na vývodu RXD kmitočtem 16x vyšším než je přenosová rychlost. Pokud je linka nečinná, je na ní logická 1. Po detekování START bitu (logická 0) jsou následující bity ukládány do přijímacího posuvného registru a odtud poté do registru UDR a současně se nastaví příznak RXC (příjem

kompletní). Jestliže nedojde k přečtení UDR dříve než se přepíše dalším přijatým bytem, nastaví se příznak OR (ztráta znaku) [12].

3.5 Paměť pro data

Pokud bude zařízení napájeno, ale nebude zapnutý program na počítači nebo nebude počítač připojen vůbec, budou se data ukládat do paměti, odkud si je bude moci počítač načíst po odeslání pokynu mikrokontroléru.

Nabízí se možnost využít paměť SRAM v mikrokontroléru, problémem je to, že tato paměť ztrácí uložená data pokud není napájena. Pokud by byl teploměr využíván pro záznam teploty mimo dosah PC při napájení např. z nabíječky na mobilní telefon, došlo by při přepojování k PC ke ztrátě dat.

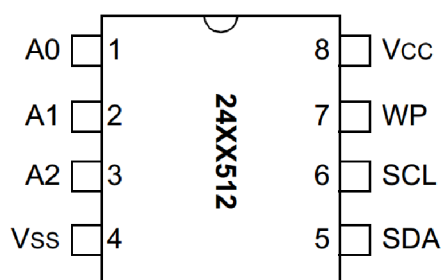
Další možností je využití paměti EEPROM integrované v mikrokontroléru, ta má však kapacitu pouze 512 B, což je při potřebě 2 B na jeden údaj o teplotě málo. Proto byla zvolena externí paměť. Zde máme opět více možností. Můžeme použít paměť typu Flash nebo EEPROM. Paměti EEPROM mají větší životnost i díky možnosti zápisu po bytech, paměť Flash umožňuje pouze zápis po blocích, což při omezeném počtu zápisů a zapisování malých objemů dat výrazně snižuje životnost. Flash paměť je rychlejší a má obvykle větší kapacitu, to však není pro toto zařízení důležité.

Nejlepší volbou by byla paměť FRAM, která kombinuje výhody pamětí RAM a ROM (non-volatilní, s téměř neomezeným počtem zápisů), problémem je horší dostupnost této paměti na našem trhu, vyšší cena a nižší kapacita. Z těchto důvodů byla zvolena paměť EEPROM, konkrétně model 24LC512 od firmy Microchip.

3.5.1 Popis paměti

Tato paměť má kapacitu 512 kb, tedy 64 kB, což při objemu 2 B na jeden údaj dává možnost zapsat více než 30000 hodnot. Dle informací výrobce umožňuje až 1000000 cyklů vymazání/zápis dat. Napájecí napětí je v rozmezí 2,5 – 5 V.

S mikrokontrolérem paměť komunikuje pomocí rozhraní I²C (v mikrokontroléru je toto rozhraní označeno TWI, dále v textu je proto použito toto označení), přes vývody označené SDA (data) a SCL (hodinový signál), viz 3.5.2. Další možností je zapojení až 8 zařízení k této sběrnici a jejich adresování pomocí adresy napevno nastavené na vývodech A0, A1 a A2. Funkce vývodu WP (write protect) je taková, že při připojení na Vcc dojde k zablokování možnosti zápisu [16].



Obr. 3.6: Pouzdro paměti [16]

3.5.2 Komunikace s pamětí

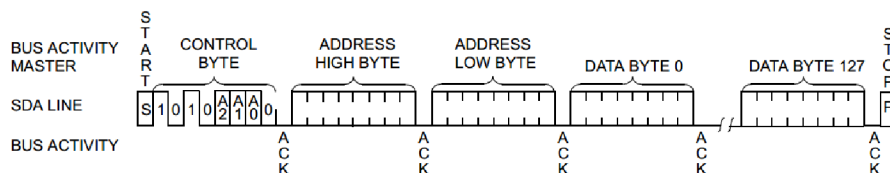
Jak již bylo zmíněno výše, paměť komunikuje pomocí sběrnice TWI. Ta využívá dvou vodičů, datový (SDA) a hodinový (SCL). K oběma vodičům je nutné připojit pull-up rezistory, zde jsou použity hodnoty 4,7 k Ω . Při komunikaci je mikrokontrolér zařízením typu master a generuje tedy hodinový signál, paměť je typu slave. Obě zařízením však mohou pracovat jako vysílač i jako přijímač.

Komunikace může začít, pokud není linka obsazena, to je indikováno logickou 1 na SCL i SDA. Poté vysílač vyšle start bit jako sestupnou hranu na SDA při SCL v logické 1. Poté následují samotná data, přenos je ukončen náběžnou hranou na SDA při SCL v 1.

Data mají následující formát: nejprve se odesílá tzv. kontrolní byte, který obsahuje sekvenci 1010, poté adresu slave zařízení se kterým chce master komunikovat a poté bit určující čtení (1) nebo zápis (0). Po přenesení kontrolního bytu odpovídá slave zařízení Acknowledge (dále jen ACK) bitem při devátém hodinovém cyklu. Další data jsou závislá na druhu komunikace.

Zápis

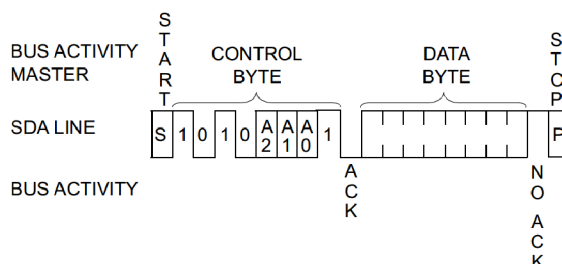
Po START bitu následuje kontrolní byte, který má na konci 0 jako zápis. Master poté vysílá dvou bytovou adresu, do které se má zapisovat, pak následují samotná data, 1 až 127 bytů ukončená STOP bitem. Po každém obdržení bytu odesílá slave ACK bit.



Obr. 3.7: Zápis dat [16]

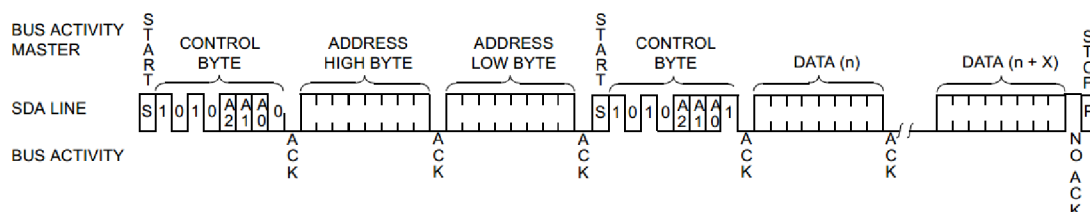
Čtení

Pro čtení máme dvě možnosti. Buď čtení z aktuální pozice, kdy po START bitu následuje kontrolní byte na konci s jedničkou pro čtení. Slave vyšle ACK bit a poté odešle byte z aktuální adresy. Master neodesílá ACK, pouze ukončí spojení STOP bitem.



Obr. 3.8: Čtení dat z aktuální adresy [16]

Druhou možností je čtení z námi zvolené adresy. Nastavení adresy se provede pomocí zápisu: START bit, kontrolní byte s 0 pro zápis a adresa ze které se má číst. Poté master znovu odešle START bit, kontrolní byte s 1 pro čtení a poté slave odesílá data z nastavené adresy. Po každém bytu master odesílá ACK, pokud chce spojení ukončit odešle STOP bit [16].



Obr. 3.9: Čtení dat z nastavené adresy [16]

3.6 LCD displej

Oproti zadání jsem k zařízení přidal i LCD displej, aby mohl teploměr zobrazovat data i bez spuštění obslužného programu na počítači. Tento displej je na samostatné desce a je k hlavní desce s mikrokontrolérem připojen 10 žilovým kabelem s konektory MLW.

Pro zobrazení teploty ve tvaru $-xx,x$ °C nebo xxx,x °C postačí 8 znaků, z důvodu možnosti budoucího rozšíření zařízení např. o další čidla jsem se rozhodl použít displej se 16x2 znaky, konkrétně model WINSTAR WH1602A-TTI-ET.

3.6.1 Popis displeje

Jedná se o alfanumerický displej s řadičem HD44780. Má 16 vývodů, jejich funkce je uvedena v tabulce 3.2. Tento řadič umožňuje přenos dat po 8 nebo 4 linkách. Ve druhém případě se využívají pouze vývody DB4 až DB7, po kterých se data odesílají rozdělená na poloviny. Já jsem zvolil 4 vodičové zapojení z důvodu ušetření vývodů mikrokontroléru.

Displej se nachází na samostatné desce, která je k hlavní desce s mikrokontrolérem připojena plochým 10žilovým kabelem s konektory PFL10 na kabelu a MLW10 na DPS. Dvě žíly kabelu jsou použity pro napájení, ostatní jsou připojeny na PORTA mikrokontroléru.

Vstupní napětí displeje je na vývod U_{cc} přivedeno přes tranzistor, který je spínán mikrokontrolérem, takže je možné displej programově vypnout. Stejným způsobem je možné ovládat podsvícení na vývodu A. Vývod U_o slouží pro nastavení kontrastu, je k němu připojen trimr s hodnotou 5 k Ω . Změnou tohoto odporu měníme napětí na U_o a tím i kontrast displeje. Nevyužité datové vývody DB0 – DB3 jsou připojeny na zem. Vývod R/\overline{W} určuje jestli budeme data číst nebo zapisovat. V praxi je však čtení z displeje zbytečné, proto je tento vývod připojen na zem, což znamená zápis. Ostatní vývody jsou připojeny přímo na port mikrokontroléru.

Tab. 3.2: Vývody displeje

Číslo vývodu	Signál	Popis
1	GND	Zem
2	U_{cc}	Napájecí napětí
3	U_o	Nastavení kontrastu
4	RS	Příkaz (0) nebo data (1)
5	R/\overline{W}	Čtení nebo zápis
6	E	Enable
7 – 14	DB0 – DB7	Datové bity
15	A	Anoda podsvětlovací LED
16	K	Katoda podsvětlovací LED

Displej disponuje dvěma typy paměti:

- DD RAM – obsahuje znaky, které jsou zobrazovány na displeji. Pro displej 16x2 znaků jsou adresovány jako 00H – 0FH pro první řádek a 40H – 4FH pro druhý řádek,
- CG RAM – umožňuje uživateli definovat až 8 vlastních znaků, např. znaky s diakritikou. Tyto znaky jsou adresovány 00H až 07H.

Jednotlivé znaky jsou matice 5x7 bodů. Tyto znaky zobrazujeme tak, že na adresu odpovídající požadované poloze na displeji odešleme jejich ASCII kód. Vlastní znaky definujeme tak, že do CG RAM odešleme na vybranou adresu 7x byte, v každém bude pět spodních bitů reprezentovat jeden řádek znaku, kdy jedničky znamenají rozsvícení bodu [17].

3.6.2 Komunikace s displejem

Nejprve je nutné provést inicializaci řadiče. Po zapnutí napájení se zvolí režim komunikace (8 nebo 4 datové bity) a počet řádků. Poté se smaže obsah displeje a displej se zapne. Dále je možné zvolit chování kurzoru (zobrazení, blikání, směr posunu).

Zápis dat (příkazu) probíhá tak, že na vývodu RS nastavíme zda jde o data (1) nebo příkaz (0). Poté se aktivuje vývod E a na DB7 až DB4 se přivede horní 4 bity dat nebo příkazu a potvrdí se deaktivací E. Stejně se přenesou dolní 4 bity. Na obrázku 3.10 je seznam příkazů pro ovládání displeje.

Data/příkaz	RS	Data								Čas provedení
		7	6	5	4	3	2	1	0	
Vymaž displej	0	0	0	0	0	0	0	0	1	1,64 ms
Návrat na začátek	0	0	0	0	0	0	0	1	x	1,64 ms
Volba režimu	0	0	0	0	0	0	1	I/D	S	40 μs
Zapni/vypni displej	0	0	0	0	0	1	R/L	C	B	40 μs
Posun kurzoru	0	0	0	0	1	S/C	0	x	x	40 μs
Nastavení komunikace	0	0	0	1	DL	N		x	x	40 μs
Nastavení adresy CG RAM	0	0	1	adresa CG RAM						40 μs
Nastavení adresy DD RAM	1	1	adresa DD RAM						40 μs	
Zápis dat do zvolené paměti	1	data pro zápis do paměti						40 μs		

Obr. 3.10: Seznam příkazů pro ovládání LCD [18]

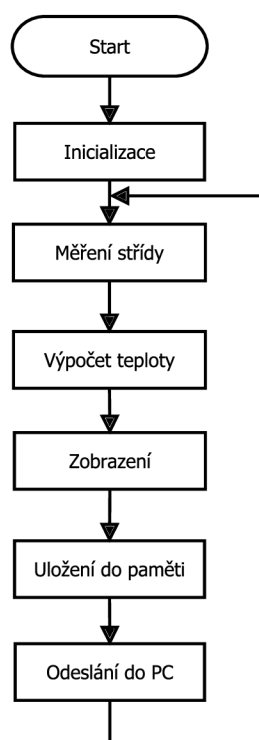
4 PROGRAMOVÉ ŘEŠENÍ

Programové řešení tohoto projektu se skládá ze dvou částí – programu pro mikrokontrolér, který řídí teploměr a zajišťuje jeho základní funkce jako je měření teploty a komunikace s PC. Druhou částí je aplikace pro PC s grafickým rozhraním, která zobrazuje výsledky měření a také slouží pro nastavení teploměru. Podrobný popis těchto programů najdete v této kapitole.

Hlavním zdrojem informací při vypracování této části práce byla internetová fóra avrfraks.net a stackoverflow.com, kde byla hledána řešení na konkrétní problémy, které se při práci objevily, proto zde nejsou uvedeny žádné odkazy na literaturu.

4.1 Program pro mikrokontrolér

Tento program je napsán v jazyce Assembler ve vývojovém prostředí WinAVR, které je dodáváno s programátorem popsaným výše. Vývojové prostředí obsahuje šablony pro většinu mikrokontrolérů firmy Atmel a díky nápovědě IntelliSense umožňuje snadnou práci s registry konkrétního modelu mikrokontroléru. Dále obsahuje debugger, ve kterém je možné simulovat vyvíjený program včetně zobrazení a změny obsahu všech registrů.



Obr. 4.1: Zjednodušený vývojový diagram programu pro mikrokontrolér

Nyní už k vlastnímu programu. Po úvodní inicializaci LCD, rozhraní USART a TWI a načtení údajů o nastavení z paměti běží hlavní program v nekonečné smyčce, která se skládá z měření střídy, jejího přepočtu na teplotu, porovnání s maximální a minimální teplotou a případné přepsání těchto hodnot a následně dle zvoleného nastavení pokračuje zobrazením teploty na displeji ve °C nebo °F, odesláním naměřené teploty do PC nebo uložení do paměti.

Zároveň se očekává přerušení od příjmu dat na rozhraní USART, která obsahují pokyny a nastavení z aplikace pro PC a také přerušení od časovače, který počítá nastavený časový interval pro ukládání teploty.

Vývojový diagram na obrázku 4.1 zobrazuje všechny bloky tohoto programu za sebou, ve skutečnosti jsou některé spouštěny přerušením a některé jsou vynechány v závislosti na nastavení, proto berte tento diagram pouze jako orientační k ilustraci funkcí programu.

Nyní se podíváme na jednotlivé části podrobněji.

4.1.1 Měření teploty

Jako teplotní čidlo je použit senzor SMT 160, který převádí teplotu na střidu obdélníkového signálu. Výstup tohoto čidla je připojen přímo na vstup mikrokontroléru PB4. V programu poběží cyklus s 30000 opakováními (pro vyšší přesnost), kdy bude při každém průběhu cyklem měřena logická úroveň na tomto vstupu. Pokud bude úroveň H, dojde ke zvýšení 16bitového počítadla, ve kterém bude po skončení měření hodnota střídy v rozsahu 0 – 30000.

```
SBIC PINB, 4           ;POKUD L, PRESKOCI ZVYSENI JEDNICEK
ADIW R31:R30, 1       ;POCET JEDNICEK
ADIW R29:R28, 1       ;POCET MERENI
```

Výpočet teploty

Teplotní rozsah čidla je –45 až 130 °C, pro uložení teploty budou použity dva byty, jeden pro uložení celé části teploty, ve druhém bude na dolních čtyřech bitech jedno desetinné místo a na MSB bude indikace záporného výsledku. Po výpočtu budou výsledky uloženy do proměnných v SRAM s názvem TEPL a TEPL2.

Pokud z rovnice 3.1, která určuje závislost střídy signálu na teplotě, vyjádříme teplotu a jednotlivé koeficienty vynásobíme 30000, kvůli počtu vzorků, dostaneme následující rovnici:

$$t = \frac{D.C. - 9600}{141}. \quad (4.1)$$

Kde t je teplota a $D.C.$ je střída signálu.

Nejprve je tedy nutné od naměřené hodnoty střídý odečíst 9600. Jelikož se jedná o dvě 16bitová čísla, je k tomu použit příkaz `SUB16` z balíčku vícebitových aritmetických operací `A32.inc`, který je součástí vývojového prostředí. Pokud je hodnota střídý menší než 9600, výsledek bude záporný, tedy naměřená teplota je záporná. Záporný výsledek je detekován, mínus je uloženo do proměnné `TEPL2` do 7. bitu a výsledek výpočtu je převeden na kladné číslo příkazem `COM`.

Nyní je třeba toto číslo vydělit 141. Jelikož mikrokontrolér neobsahuje instrukci pro dělení, je použit algoritmus postupného odečítání, kdy je po každém odečtení po kterém není výsledek záporný zvýšeno počítadlo, ve kterém je pak uložen výsledek po dělení, který je pak přesunut do proměnné s názvem `TEPL`. Zbytek po dělení je poté vynásoben deseti a opět postupným odečítáním vydělen 141 pro získání jednoho desetinného místa, které je namaskováno na spodní čtyři bity proměnné `TEPL2`.

Pro zvýšení přesnosti měření a zamezení neustálého kolísání výsledku na místě desetín je měření střídý opakováno $8\times$ a z těchto hodnot střídý je spočítán průměr.

4.1.2 Zobrazení teploty na displeji

Pro práci s displejem je využívána knihovna `LCD_M08.inc`, kterou mám k dispozici od střední školy a tudíž bylo zbytečné vytvářet vlastní program. Tato knihovna obsahuje makra a podprogramy pro veškeré ovládání jedno, dvou a čtyřřádkových LCD. Po úvodní inicializace provedené makrem `INIT_LCD` se zobrazení dat provádí uložení pozice na displeji do registru `R16` a voláním podprogramu `RADX` kde `X` je číslo řádku, na který chceme zapisovat a následně zapsáním ASCII kódu požadovaného znaku do registru `R16` a zavoláním podprogramu `VYSDATA`.

Na prvním řádku displeje je zobrazena aktuální teplota, na druhém se pak bude střídát maximální a minimální teplota od zapnutí přístroje.

Před zobrazením je nutné teplotu převést do BCD kódu, k čemuž slouží podprogram `BINBCD`, dále je nutné detekovat znaménko mínus a odstranit případné přebytečné nuly na začátku čísla.

V aplikaci na PC bude také možné volit mezi zobrazováním ve $^{\circ}\text{C}$ nebo $^{\circ}\text{F}$. Ve druhém případě je nutné vypočítanou hodnotu teploty převést na Fahrenheitovu stupnici pomocí následujícího vztahu:

$$t [^{\circ}\text{F}] = \frac{9}{5} \cdot t [^{\circ}\text{C}] - 32. \quad (4.2)$$

V nastavení na PC bude možné též vypnout displej nebo jeho podsvícení, například pro úsporu energie nebo pokud bude teploměr sloužit pouze pro záznam teploty do paměti.

4.1.3 Uložení dat do paměti

Jak již bylo zmíněno výše, použitá paměť je typu EEPROM, má kapacitu 64 kB a pro komunikaci používá sběrnici TWI. Jednotlivé buňky v paměti jsou adresovány dvěma byty, přičemž prvních 256 bytů je použito pro uložení řídicích informací jako údaje o nastavení teploměru a adresa, na které se má pokračovat se zápisem. Vlastní naměřená data budou ukládána od adresy 0x01 v horním bytu adresy.

Ukládání dat je prováděno v časových intervalech, které je možné nastavit v ovládací aplikaci na PC. Počítání intervalů má na starosti časovač 1, který vyvolává přerušování vždy po 1 sekundě a porovnává hodnotu počítadla s nastaveným intervalem, při shodě dojde k volání podprogramu pro ukládání do paměti.

Inicializace rozhraní

Inicializace rozhraní TWI se skládá pouze z nastavení předděličky kmitočtu, která zde není použita a nastavení frekvence hodinového signálu na vodiči SCL. To se provede podle vzorce

$$f_{SCL} = \frac{f_{clk}}{16 + 2 \cdot TWBR \cdot 4^{TWPS}}, \quad (4.3)$$

kde f_{SCL} je frekvence na SCL, f_{clk} je pracovní frekvence mikrokontroléru, TWPS je předdělička a TWBR je registr pro nastavení rychlosti TWI.

V tomto přístroji je použita frekvence 100 kHz, tedy hodnota $TWBR = 47$.

Práce s rozhraním TWI

Ovládání rozhraní TWI je velmi jednoduché. Využívá se zde dvou registrů, TWCR (řídicí registr) a TWDR (datový registr). Význam jednotlivých bitů řídicího registru,

7	6	5	4	3	2	1	0
TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE

Obr. 4.2: Registr TWCR pro řízení rozhraní TWI

které jsou použity v tomto programu:

- TWINT – jednička v tomto bitu indikuje dokončení operace a zároveň zápisem jedničky dojde ke spuštění operace na rozhraní TWI.
- TWEA – používá se při čtení více bytů dat, pokud je nastaven, generuje se NACK bit signalizující požadavek na další data.
- TWSTA – generování START bitu.
- TWSTO – generování STOP bitu.

- TWWC – příznak kolize na lince.
- TWEN – povolení rozhraní TWI.
- TWIE – povolení přerušení.

Příklad zápisu jednoho bytu dat do paměti přes rozhraní TWI si můžete prohlédnout v příloze F.

Práce s časovačem

Časovač vyvolává přerušení po 1 sekundě. Pro tento účel je použit čítač/časovač 1. Tento časovač je 16bitový a pracuje v módu CTC (Clear Timer on Compare Match). Přerušení je vyvoláno při shodě čítacího registru TCNT1 s registrem OCR1A. Pro správnou funkci je tedy nutné nastavit do OCR1A správnou hodnotu. Pro výpočet této hodnoty slouží následující rovnice:

$$f_{OCR1A} = \frac{f_{clk}}{2 \cdot N \cdot (1 + OCR1A)}, \quad (4.4)$$

kde f_{OCR1A} je frekvence přerušení, převrácená to hodnota požadovaného intervalu, f_{clk} je pracovní frekvence mikrokontroléru, N je předdělička a OCR1A je registr pro nastavení časovače v módu CTC.

Intervalu 1 sekunda při frekvenci krystalu 11,0592 MHz a použité předděličky 1024 odpovídá hodnota v 16bitovém registru OCR1A 10799. Dále je nutné nastavit požadovaný mód pomocí registrů TCCR1A a TCCR1B a povolit přerušení v registru TIMSK. Při přerušení dojde k zavolání podprogramu SEKUNDA, který zajišťuje kontrolu intervalu pro ukládání, v případě rovnosti počítadla s tímto intervalem dojde k uložení aktuální teploty do paměti. Tento podprogram slouží též pro přepínání mezi zobrazováním maxima a minima na displeji.

4.1.4 Komunikace s PC

Jak již bylo zmíněno, pro komunikaci s PC je použit převodník USART – USB. Mikrokontrolér ATmega16 má zabudovaný asynchronní sériový kanál, který je v tomto programu využit.

Inicializace USART

Inicializace sériového kanálu se skládá nastavení přenosové rychlosti a volby parametrů přenosu. Nastavení rychlosti se provádí 16bitovým registrem UBRR, jeho hodnota se spočítá ze vzorce:

$$UBRR = \frac{f_{clk}}{16 \cdot PR} - 1, \quad (4.5)$$

kde f_{clk} je pracovní frekvence mikrokontroléru a PR je přenosová rychlost.

Při krystalu s frekvencí 11,0592 MHz, bez předděličky a rychlosti 230400 Bd je hodnota TWBR 2.

Nastavení parametrů přenosu se provede pomocí registrů UCSRA, UCSRB a UCRSC. V tomto programu je v registru UCSRB nastaveno povolení příjmu a vysílání a povolení přerušování při příjmu, v registru UCSRC je pak nastaven asynchronní mód, 1 stop bit, 8 datových bitů a žádná parita.

Odesílání dat má na starosti podprogram ODESLANI který nejprve kontroluje příznak UDRE v registru UCSRA, který určuje, zda je možné vysílat a poté jednoduše přesune data do registru UDR, čímž zahájí přenos.

LDS R16, TEPL
OUT UDR, R16

Příjem dat je zajištěn pomocí přerušování při příjmu, které spustí podprogram PRIJEM. V něm se vyhodnotí přijatý příkaz a je spuštěn požadovaný podprogram.

Komunikační protokol

Počítač odesílá do mikrokontroléru bajtové příkazy. Pokud je na PC zapnut ovládací program, je do mikrokontroléru odeslán příkaz 0xFF indikující zahájení komunikace. Mikrokontrolér odpoví odesláním bytu s aktuálním nastavením, aby toto mohlo být zobrazeno v ovládací aplikaci a poté pokračuje odesláním aktuální teploty, vždy po skončení výpočtu. Každá hodnota teploty se skládá ze dvou bytů. Při vypnutí aplikace na PC je vyslán povel 0x5A pro ukončení odesílání dat z mikrokontroléru, měření však pokračuje dál.

Dalším příkazem je požadavek od PC na načtení dat z paměti, který má hodnotu 0x50. Po jeho přijetí je nejprve je vyslán vyslán kód 0xFF, který symbolizuje odesílání dat z paměti a poté následují samotná data. Spojení je ukončeno kódem 0xFE.

Tab. 4.1: Příkazy od PC

Hodnota příkazu	Popis funkce
0xFF	Zapnutí programu, zahájení přenosu
0x50	Načtení dat z paměti
0x55	Změna nastavení
0x5A	Ukončení aplikace, konec přenosu
0x5E	Vymazání paměti

Pro změnu nastavení teploměru slouží příkaz 0x55. Po něm následuje další byte, který obsahuje informace o nastavení. Význam jednotlivých bitů v tomto bytu je

v tabulce 4.2. Mikrokontrolér tento byte uloží do externí paměti i do paměti SRAM a poté provede příslušné změny. Při běhu hlavního programu je tento byte načítán z SRAM a na základě jeho hodnoty je prováděno rozhodování (volba jednotky, ukládání, zobrazování). Posledním z použitých příkazů je 0x5E pro vymazání paměti. Po jeho přijetí nedojde k fyzickému vymazání paměti, ale pouze k nastavení aktuální adresy na začátek paměti.

Tab. 4.2: Význam bitů v nastavovacím bytu

Bit	Popis funkce
7	Výběr jednotky
6	Zapnutí/vypnutí displeje
5	Zapnutí/vypnutí podsvícení displeje
4	Rezervováno
3	Zapnutí/vypnutí ukládání do paměti
2, 1, 0	Volba intervalu pro ukládání do paměti

4.2 Aplikace pro PC

Ovládací aplikace pro PC je vytvořena ve vývojovém prostředí Microsoft Visual Studio 2012 v jazyce C#. Jedná se o aplikaci s grafickým rozhraním založenou na technologii WPF (Windows Presentation Foundation) a je vytvořena pro .NET Framework 4.5. Aplikace slouží k zobrazení naměřených hodnot a nastavení teploty měru.

Pro správnou funkci aplikace je nutné nejprve nainstalovat ovladače od USART – USB převodníku, které jsou dostupné na stránkách výrobce FTDI. Tento převodník je poté možné ovládat jako virtuální sériový port nebo pomocí knihovny FTD2XX.dll, která obsahuje metody a události pro přímé ovládání převodníku. V této aplikaci je použita druhá možnost, neboť nabízí širší možnosti nastavení a v ledasčem usnadňuje a zpřehledňuje práci.

Kromě této knihovny se aplikace skládá ze tří oken a k nim patřících tříd, které obsahují metody pro ovládání těchto oken a ze třídy Usb.cs, ve které jsou metody pro ovládání rozhraní USB a různé další funkce.

4.2.1 Inicializace rozhraní USB

Nejprve je nutné vytvořit instanci objektu třídy FTDI, přes kterou se volají všechny metody pro práci s převodníkem a jeho nastavení. Každá tato metoda vrací proměnnou typu FTSTATUS, ve které je uložen stav provedené operace (FT_OK pro úspěšné

provedení, jinak typ chyby). Proto je vytvořena ještě proměnná tohoto typu, pomocí které může být sledován stav prováděných operací a můžou být ošetřeny chyby.

Vlastní inicializace, která je provedena po spuštění aplikace, se skládá ze zjištění připojených zařízení a otevření požadovaného pomocí metody `OpenBySerialNumber`. Poté je nastavena přenosová rychlost a parametry přenosu stejně jako na straně mikrokontroléru.

4.2.2 Příjem dat z teploměru

Čtení dat z USB je zajištěno pomocí třídy `Background Worker`. Ta, jak název napovídá, vytváří vlákno pracující na pozadí a spouští metodu pro čtení `Prijem` vždy při příjmu dat, což je signalizováno událostí `FT_EVENT_RXCHAR`. Po příjmu obou bytů jsou data přepočteny na číslo typu `double`, výsledná teplota je zobrazena na hlavním okně a přidána do kolekce `ObservableCollection` s prvky typu `<KeyValuePair<DateTime, double>` spolu s aktuálním časem. Tato kolekce posléze slouží jako zdroj dat pro graf.

Ukázka příjmu dat:

```
ftStatus = teplomer.Read(rbuff, 2, ref prijatych);
if (ftStatus != FTDI.FT_STATUS.FT_OK)
{
    chybaPrenosu();
    break;
}
```

Načítání dat z paměti teploměru je spuštěno v metodě `Prijem`, všechna data jsou přijata najednou, zobrazena v dalším okně v `ListBoxu` a uložena do další kolekce `ObservableCollection` sloužící opět jako zdroj dat pro graf. Zde je nevýhodou, že na osu X nemůžeme vynášet přesnou dobu změřené teploty, protože samostatně měřící teploměr nemá přístup k aktuálnímu času. Na této ose je tedy zobrazen čas od začátku měření, který je vypočítaný z intervalu ukládání.

Kreslení grafů

Pro kreslení grafů je zde použito komponenty `Chart` z balíčku WPF toolkit. Tato komponenta umožňuje výběr z mnoha typů grafů, obsahuje několik typů os a nabízí velké množství přizpůsobení vzhledu výsledného grafu. V našem případě je do komponenty `Chart` přidán objekt typu `LineChart`, tedy čárový graf. Tomuto objektu se poté jen přiřadí k vlastnosti `ItemsSource` kolekce s prvky typu `KeyValuePair`, ve které jsou uložena data k zobrazení.

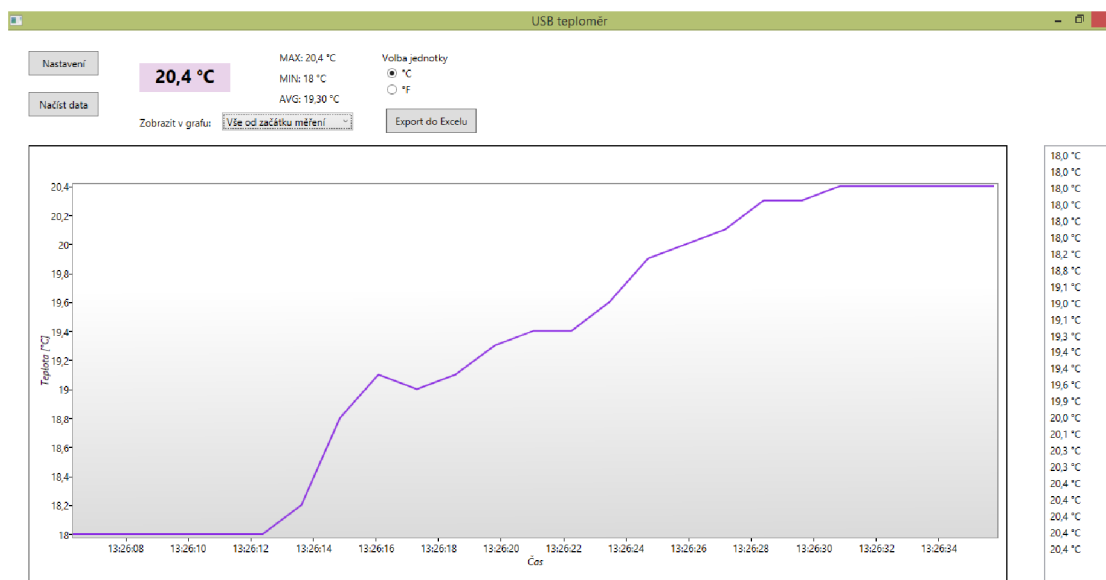
V aplikaci je též možnost výběru oblasti dat k zobrazení (např. poslední minuta). Toho je dosaženo změnou minima na ose X a tím vlastně k oříznutí grafu.

4.2.3 Popis uživatelského rozhraní

Aplikace se skládá ze tří oken. Na hlavním okně se nachází komponenta TextBlock pro zobrazení aktuální teploty, plocha pro vykreslování průběhu teploty, ListBox se všemi naměřenými hodnotami, tři TextBlocky se zobrazením maximální, minimální a průměrné teploty, ComboBox pro výběr oblasti zobrazené v grafu, RadioButtony pro volbu jednotky, tlačítko pro export dat do Excelu a ovládací tlačítka.

Tlačítko „Nastavení“ otevře další okno s možnostmi nastavení teploměru. Zde se nachází CheckBoxy pro zapnutí/vypnutí displeje, jeho podsvícení a ukládání dat do paměti. Dalším prvkem je ComboBox pro výběr intervalu ukládání, možné je zvolit z předdefinovaných intervalů 1 sekunda, 30 sekund, 1 minuta, 30 minut, 1 hodina, 6 hodin a 12 hodin. K výběru jednotky pro zobrazení teploty na displeji slouží dva RadioButtony. Posledními prvky jsou čtyři tlačítka. Tlačítko „Vymazat paměť“ slouží k vymazání paměti v teploměru, stisknutím „OK“ se údaje o nastavení odeslou a okno se zavře, tlačítkem „Použít“ dojde také k odeslání nastavení, ale okno zůstane otevřené a nakonec stisknutím „Zrušit“ se okno zavře bez odeslání změn.

Dalším tlačítkem na hlavním okně je „Načíst data“, po jehož stisku dojde k odeslání pokynu pro načtení dat, otevře se nové okno a po příjmu veškerých dat se v něm zobrazí tabulka s daty a graf.



Obr. 4.3: Screenshot hlavního okna aplikace

5 KONSTRUKCE

Zařízení se skládá ze dvou desek plošných spojů. První deska je hlavní, na ní se nachází mikrokontrolér, paměť a převodník USB – USART. Na druhé desce je pak displej s podpůrnými součástkami. Obě desky jsou propojeny plochým kabelem. Schémata i návrhy DPS byly vytvořeny v programu Eagle. Schémata i osazovací plány naleznete v příloze. Podle těchto návrhů byly desky vyrobeny ve školní laboratoři tavním zaměstnancem. Všechny součástky mají drátové vývody, jejich pájení tak bylo velmi snadné.

Obě vyrobené a osazené desky jsou umístěny v černé kovové krabičce o rozměrech 66 x 102 x 146 mm (výška x šířka délka). Desky jsou uchyceny šrouby na distančních sloupcích. Na přední straně krabičky je otvor, ve kterém se nachází displej, upevněný na DPS pomocí kolíkové lišty se zahnutím 90°. Na zdaní straně je konektor USB B, který je součástí modulu převodníku a slouží tedy pro připojení k počítači a konektor USB A pro připojení čidla.

Čidlo je umístěno na čtyř žilovém kabelu s délkou 1 metr. Připojení pomocí konektoru umožňuje prodloužit kabel až na 20 m, dle údajů výrobce čidla. Čidlo je i s přilehlými holými vodiči zalito do epoxidu, díky tomu je možné jej použít i pro měření teploty vodivých kapalin, např. vody.



Obr. 5.1: Fotografie výrobku

ZÁVĚR

Zadáním bakalářské práce bylo vyrobit elektronický teploměr, který bude komunikovat s počítačem přes rozhraní USB.

Prostudoval jsem problematiku elektronického měření teploty a komunikace přes USB rozhraní. Na základě získaných informací jsem navrhl zapojení teploměru s mikrokontrolérem, využívající převodník USART – USB pro komunikaci s počítačem. Také jsem navrhl samostatný modul s LCD displejem, který slouží pro zobrazení údajů o teplotě. Podle mého návrhu byly vyrobeny desky plošných spojů. Následně jsem tyto desky osadil vybranými součástkami.

Hlavní částí práce bylo programování mikrokontroléru a aplikace pro počítač. Program pro mikrokontrolér je napsán v jazyce Assembler a řídí chod celého teploměru. Měří teplotu, komunikuje s počítačem a naměřená data ukládá do paměti a zobrazuje je na displeji.

Aplikace pro PC má grafické rozhraní a zobrazuje aktuální teplotu, vynáší tuto teplotu do grafu a poskytuje přehled maximální, minimální a průměrné teploty od začátku měření. Aplikace také umožňuje načtení dat z paměti teploměru a jejich prezentaci stejně jako v případě aktuální teploty. Poslední možností této aplikace je nastavení teploměru, ať už jde o výběr jednotky pro zobrazení teploty na displeji nebo volba intervalu pro ukládání dat do paměti.

Na závěr jsem celé zařízení zabudoval do krabičky a čidlo vyvedl na kabel s konektorem, umožňujícím prodloužení tohoto kabelu a zalil je do epoxidu kvůli měření teploty ve vodivých kapalinách.

Zadání práce se mi tedy podařilo splnit, nicméně výsledný produkt není ideální. Vhodným rozšířením by bylo zabudování obvodu reálného času, aby mohl být při ukládání do paměti k údaji o teplotě uložen i údaj o času měření. Také teplotní čidlo není nejvhodnější. Čidlo vykazuje poměrně velkou setrvačnost, která je způsobena jak samotnou technologií čidla, tak způsobem prezentace výsledků, kdy je nutné opakovaně vzorkovat signál a také zalitím čidla do epoxidu, kdy vzrostl objem materiálu, který je nutné zahřát/zchladit.

Nicméně přesnost čidla je dostatečná. Byl proveden pokus s ponořením čidla do vody s ledem, teplota se ustálila přesně na 0 °C. Také bylo provedeno měření teploty varu vody, kdy teploměr ukazoval 97,1 °C. Tento pokus byl proveden v nadmořské výšce asi 535 m.n.m., čemuž by měla odpovídat teplota asi 97,8 °C. Pro plánované využití teploměru je toto čidlo dostačující.

LITERATURA

- [1] KRIEDL, M. *Měření teploty – senzory a měřicí obvody* 1. vyd. Praha: BEN – technická literatura, 2005, 240 s. ISBN 80-730-0145-4.
- [2] Automatizace.hw.cz: Teorie zpracování signálu platinových teplotních senzorů. In: [online]. [cit. 2014-11-13]. Dostupné z: <http://automatizace.hw.cz/zpracovani-signalu-platinovych-senzoru>
- [3] ĎAĎO, S. ,KREIDL, M. *Senzory a měřicí obvody*. 1. vyd. Praha: ČVUT, 1996, 315 s. ISBN 80-010-1500-9.
- [4] Hw.cz: Měření teploty – polovodičové odporové senzory teploty. In: [online]. [cit. 2014-11-13]. Dostupné z: <http://www.hw.cz/teorie-a-praxe/dokumentace/mereni-teploty-polovodicove-odporove-senzory-teploty.html>
- [5] KAINKA, B. *USB – Měření, řízení a regulace pomocí sběrnice USB*. 1. vyd. Praha: BEN – technická literatura, 2002, 247 s. ISBN 80-730-0073-3.
- [6] Wikipedie: Universal Serial Bus. [online]. [cit. 2014-11-20]. Dostupné z: http://cs.wikipedia.org/wiki/Universal_Serial_Bus
- [7] L-com Global Connectivity: USB (Universal Serial Bus) Tutorial. [online]. [cit. 2014-11-24]. Dostupné z: <http://www.l-com.com/content/USB-Tutorial.html>
- [8] MATOUŠEK, David. *USB prakticky: s obvody FTDI*. 1. vyd. Praha: BEN – technická literatura, 2003, 270 s. ISBN 80-730-0103-9.
- [9] How stuff works: How FireWire Works. [online]. [cit. 2014-12-03]. Dostupné z: <http://computer.howstuffworks.com/firewire.htm>
- [10] Cnews.cz: ExtraHardware. [online]. [cit. 2014-11-25]. Dostupné z: <http://www.cnews.cz/thunderbolt-bleskova-prace-s-periferiemi>
- [11] Smartec Sensors: SMT160-30. [online]. [cit. 2014-12-07]. Dostupné z: <http://www.smartec-sensors.com/assets/files/pdf/sensors/SMT16030N.pdf>
- [12] MATOUŠEK, David. *Práce s mikrokontroléry Atmel AVR AT90S*. 1. vyd. Praha: BEN – technická literatura, 2003, 375 s. ISBN 80-730-0088-1.
- [13] Atmel: ATmega16. [online]. [cit. 2014-12-09]. Dostupné z: <http://www.atmel.com/images/doc2466.pdf>

- [14] WinAVR. [online]. [cit. 2014-12-01]. Dostupné z: <http://winide51.wz.cz/avr/>
- [15] FTDI: MM232R. [online]. [cit. 2014-12-07]. Dostupné z: http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_MM232R.pdf
- [16] MICROCHIP: 24AA512/24LC512/24FC512. [online]. [cit. 2014-12-07]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/21754M.pdf>
- [17] Winstar: WH1602A. [online]. [cit. 2014-12-08]. Dostupné z: http://www.winstar.com.tw/products_detail_ov.php?lang=en&ProID=21
- [18] MATOUŠEK, David. *Práce s inteligentními displeji LCD: [znakové a grafické displeje, přípravy a programy]*. 1 vyd. Praha: BEN – technická literatura, 2006, 222 s. ISBN 80-730-0121-7.

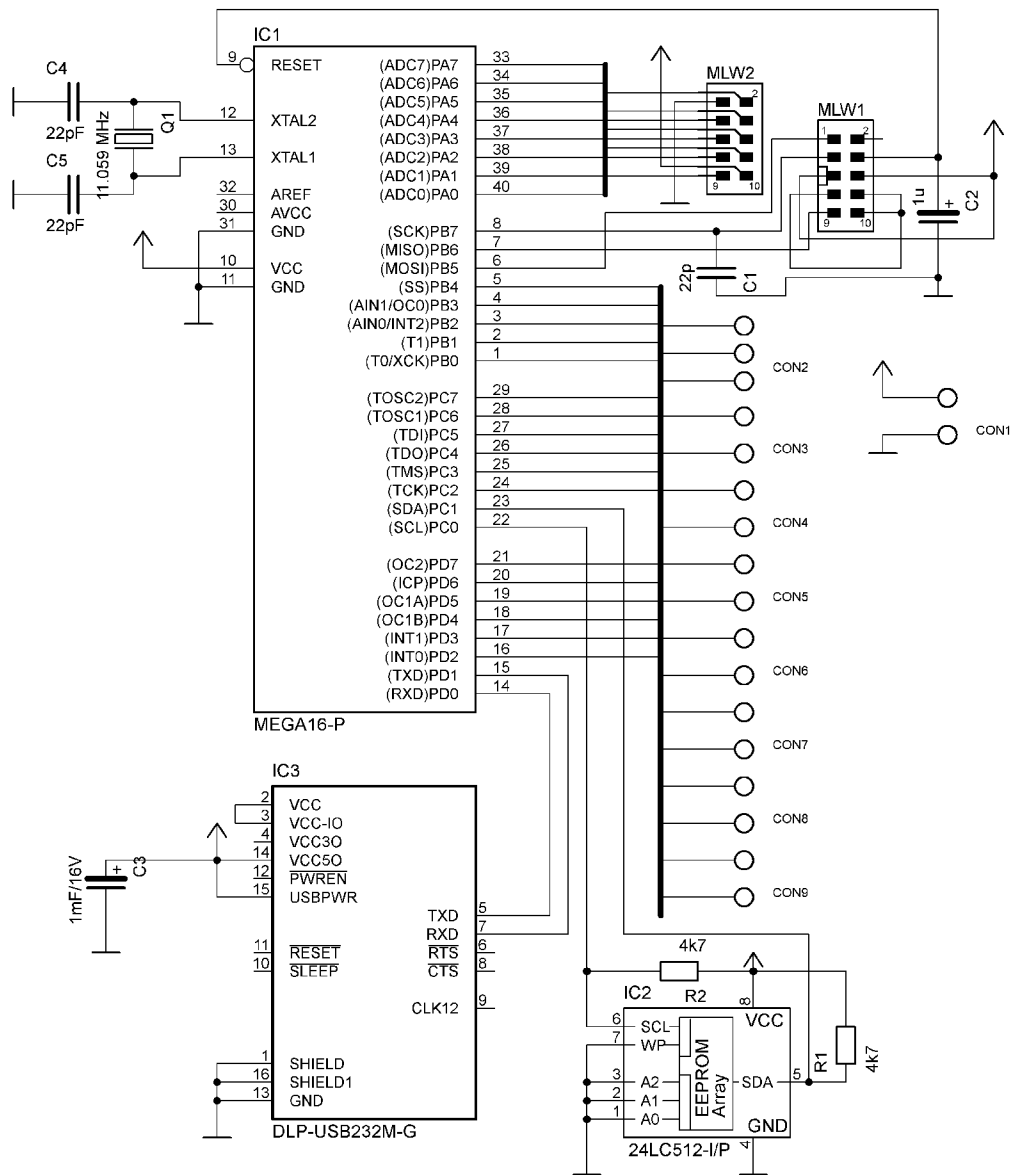
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

D.C.	Duty Cycle
DPS	Deska plošných spojů
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIFO	First In, First Out
FRAM	Ferroelectric Random Access Memory
I ² C	Inter-Integrated Circuit
NRZI	Non Return To Zero
NTC	Negative temperature coefficient
PTC	Positive temperature coefficient
RISC	Reduced Instruction Set Computer
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
TWI	Two Wire Interface
USART	Universal Synchronous / Asynchronous Receiver and Transmitter

SEZNAM PŘÍLOH

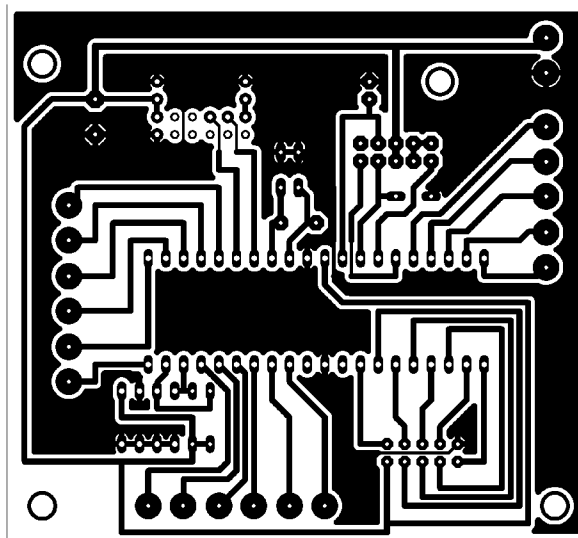
A Schéma hlavní desky	51
B Deska plošného spoje hlavní desky	52
C Schéma LCD modulu	53
D Deska plošného spoje LCD modulu	54
E Seznam součástí	55
F Příklad zápisu do paměti	56
G Fotografie výrobku a screenshoty z aplikace	57
H Obsah příloženého CD	60

A SCHÉMA HLAVNÍ DESKY

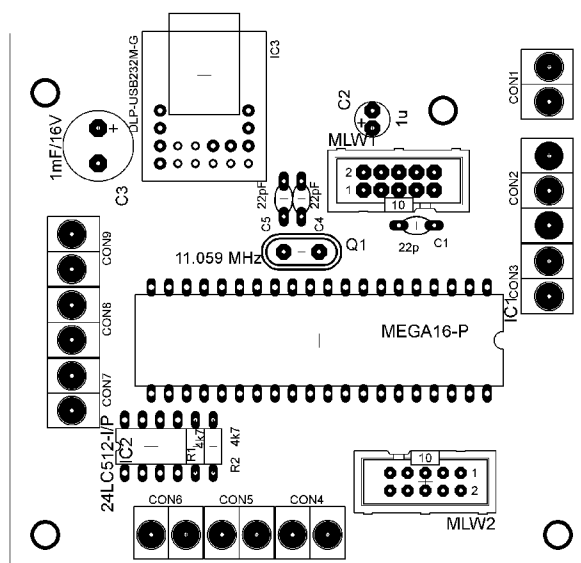


Obr. A.1: Schéma hlavní desky

B DESKA PLOŠNÉHO SPOJE HLAVNÍ DESKY

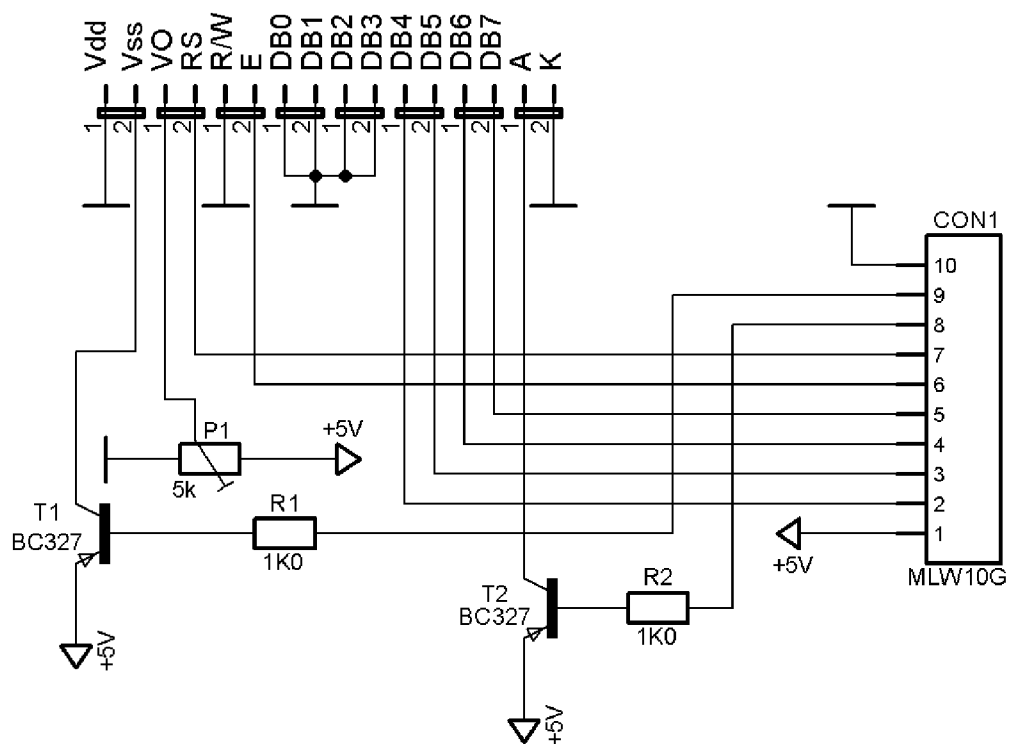


Obr. B.1: Hlavní deska, strana spojů



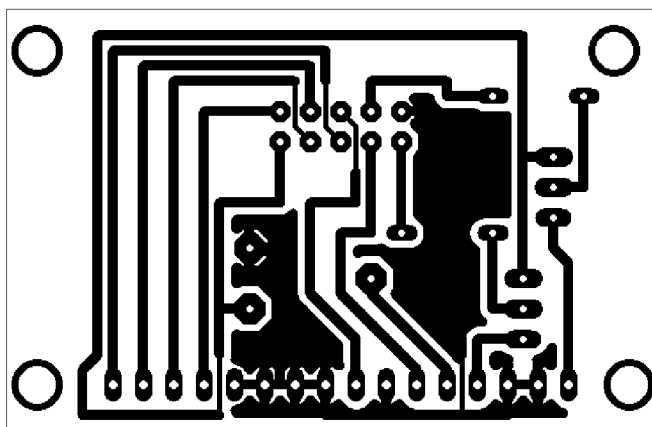
Obr. B.2: Hlavní deska, strana součástek

C SCHÉMA LCD MODULU

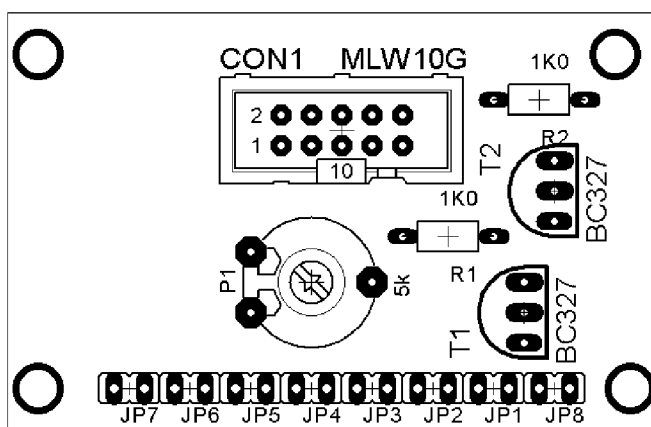


Obr. C.1: Schéma modulu displeje

D DESKA PLOŠNÉHO SPOJE LCD MODULU



Obr. D.1: Modul displeje, strana spojů



Obr. D.2: Modul displeje, strana součástek

E SEZNAM SOUČÁSTEK

Tab. E.1: Seznam součástek

Označení	Popis	Hodnota
Hlavní deska		
C1, C4, C5	Keramický kondenzátor	22 pF
C2	Elektrolytický kondenzátor	1 μ F/16 V
C3	Elektrolytický kondenzátor	1 mF/16 V
R1, R2	Rezistor	4k7
IC1	Mikrokontrolér	ATmega16
IC 2	Paměť	24LC512
IC 3	Převodník USART – USB	MM232R
MLW1, MLW 2	Konektor pro plochý kabel	–
CON1 – 9	Svorky	ARK2
–	Teplotní čidlo	SMT160-30
Modul LCD		
R1, R2	Rezistor	1k0
T1, T2	Tranzistor PNP	BC327
P1	Trimr	5k0
CON1	Konektor pro plochý kabel	MLW10
–	Kolíková lišta	14 pinů, 90°
–	Displej	Winstar WH1602A
Konstrukce		
–	Konektory pro plochý kabel	PFL10
–	Plochý kabel	–
–	Krabička	AH311
–	Konektor USB A zásuvka	–
–	Kabel s konektorem USAB A	–

F PŘÍKLAD ZÁPISU DO PAMĚTI

```
LDI R16, (1<<TWINT) | (1<<TWSTA) | (1<<TWEN) ;START BIT
OUT TWCR, R16
RCALL CEKANI
```

```
LDI R16, 0B10100000 ;KONTROLNÍ BYTE S ADRESOU ZAŘÍZENÍ
OUT TWDR, R16
LDI R16, (1<<TWINT) | (1<<TWEN) ;VYSLÁNÍ
OUT TWCR, R16
RCALL CEKANI
```

```
LDI R16, 0 ;HORNÍ BYTE ADRESY DAT V PAMĚTI
OUT TWDR, R16
LDI R16, (1<<TWINT) | (1<<TWEN) ;VYSLÁNÍ
OUT TWCR, R16
RCALL CEKANI
```

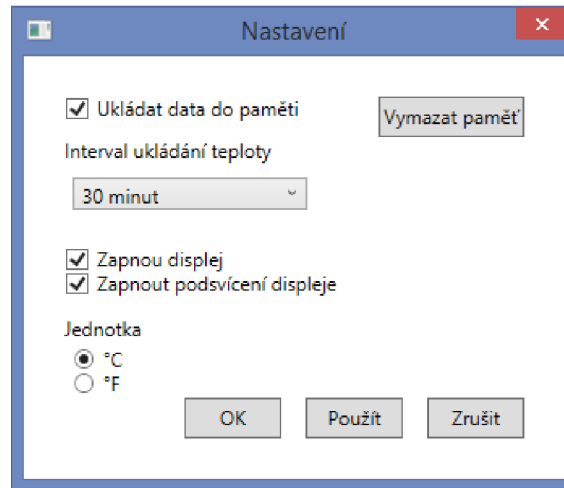
```
LDI R16, 0 ;DOLNÍ BYTE ADRESY
OUT TWDR, R16
LDI R16, (1<<TWINT) | (1<<TWEN) ;VYSLÁNÍ
OUT TWCR, R16
RCALL CEKANI
```

```
LDI R16, 65 ;DATA
OUT TWDR, R16
LDI R16, (1<<TWINT) | (1<<TWEN) ;VYSLÁNÍ
OUT TWCR, R16
RCALL CEKANI
```

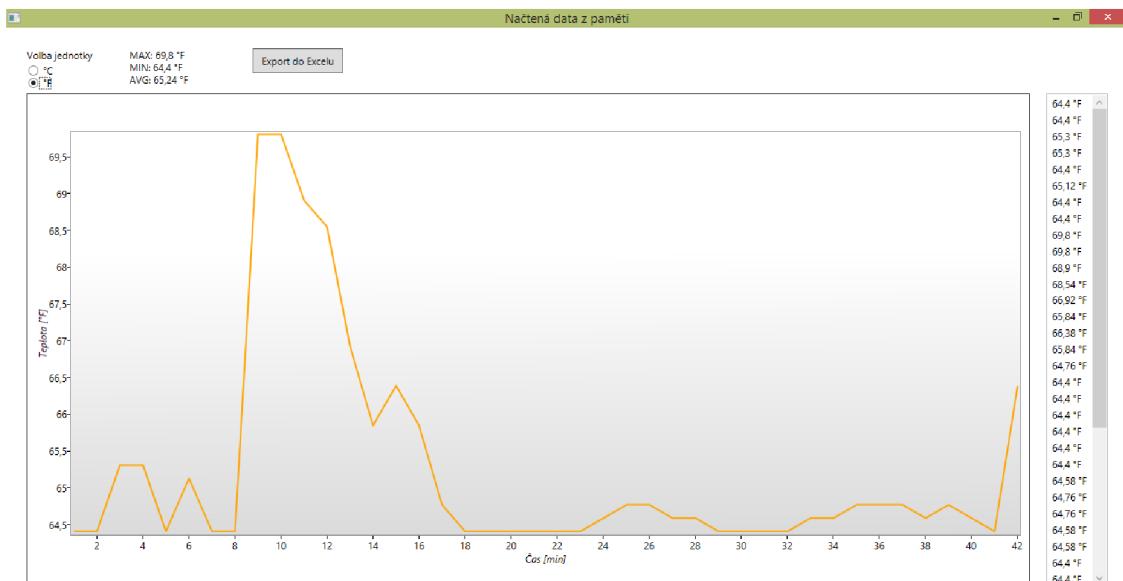
```
LDI R16, (1<<TWINT) | (1<<TWSTO) | (1<<TWEN) ;STOP
OUT TWCR, R16
RET
```

```
CEKANI:
IN R16, TWCR
SBRS R16, 7
RJMP CEKANI
RET
```

G FOTOGRAFIE VÝROBKU A SCREENSHOTS Z APLIKACE



Obr. G.1: Screenshot okna s nastavením



Obr. G.2: Screenshot okna s načtenými daty



Obr. G.3: Fotografie výrobku, pohled zepředu



Obr. G.4: Fotografie výrobku, pohled zezadu



Obr. G.5: Fotografie výrobku, čidlo

H OBSAH PŘILOŽENÉHO CD

Na přiloženém CD naleznete tyto položky:

- složka DPS, která obsahuje schémata a návrhy DPS v programu Eagle,
- složka PC obsahující zdrojové kódy k ovládací aplikaci,
- složka uC, obsahující zdrojové kódy programu pro mikrokontrolér,
- spustitelný soubor CDM v2.12.00 WHQL Certified, sloužící k instalaci ovladačů pro převodník,
- elektronická verze tohoto textu s názvem holub_BP,
- spustitelný soubor USB teploměr, finální verze aplikace pro PC,
- spustitelný soubor WPFToolkit pro instalaci balíčku pro kreslení grafů.