

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ŘÍZENÍ TEPLoty V MÍSTNOSTI

BAKALÁŘSKÁ PRÁCE

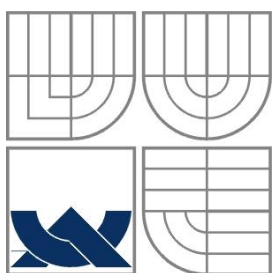
BACHELOR'S THESIS

AUTOR PRÁCE

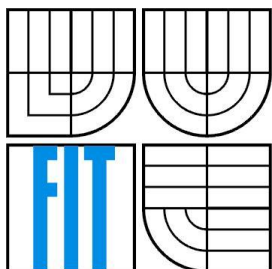
AUTHOR

ANTONÍN HRUŠKOVSKÝ

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ŘÍZENÍ TEPLOTY V MÍSTNOSTI

ROOM TEMPERATURE CONTROLLER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ANTONÍN HRUŠKOVSKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. LUKÁŠ MIČULKA

BRNO 2013

Abstrakt

Tato práce se zabývá výrobou pokojového termostatu pro regulaci teploty v domě či bytu. V textu jsou popsány typy senzorů pro měření teploty pracující na základě různých fyzikálních principů. Z těchto typů senzorů je následně vybrán jeden a ten je použit v měřící jednotce. Následně je popsán hardwarový návrh termostatu od návrhu DPS na počítači, přes výběr součástek, až po samotnou výrobu termostatu. Poté následuje popis algoritmů implementovaných na použitém mikrokontroléru MSP430, pro správnou funkci termostatu. Na straně počítače je implementován software, který slouží pro komunikaci s termostatem. Komunikace mezi počítačem a termostatem je realizována pomocí sběrnice USB.

Abstract

This bachelor thesis deals with a production of a room thermostat for regulation of temperature in a house or flat. The text describes several types of sensors for measuring temperature according to various physical principles. Afterwards, one of the types of sensors has been selected and used in a measuring unit. Subsequently, a hardware design of the thermostat has been described, including the creation of the PCB design on a computer, component selection and the thermostat production itself. This is followed by the description of algorithms implemented on the microcontroller MSP430 to ensure the proper functioning the thermostat. A software has been installed into the side of the computer used to communicate with the thermostat. The communication with the thermostat has been realized by means of an USB bus.

Klíčová slova

Mikroprocesor MSP430, termostat, teplotní čidlo DS18B20, LCD, FT232RL, USB, teplota

Keywords

Microprocessor MSP430, thermostat, temperature sensor DS18B20, LCD, FT232RL, USB, temperature

Citace

Antonín Hruškovský: Řízení teploty v místnosti, bakalářská práce, Brno, FIT VUT v Brně, 2013

Řízení teploty v místnosti

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Lukáše Mičulky. Další informace mi poskytl pan Ing. Václav Šimek během odborných konzultací. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Antonín Hruškovský
15. května 2013

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu panu Ing. Lukáši Mičulkovi za to, že mi věnoval čas a ochotně odpovídal na dotazy týkající se této práce. A to jak přes email, tak i na osobních konzultacích. Dále bych chtěl poděkovat panu Ing. Václavu Šimkovi za cenné rady a odbornou pomoc při realizaci hardwarové části bakalářské práce.

© Antonín Hruškovský, 2013

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Základní pojmy	4
2.1 Teplota	4
2.2 Teplotní stupnice	4
3 Teplotní čidla a senzory	5
3.1 Kovové odporové senzory	5
3.2 Polovodičové odporové senzory	6
3.2.1 Termistory.....	6
3.2.2 Polovodičové monokrystalické senzory	7
3.3 Monolitické PN senzory	7
3.4 Termoelektrické články	7
3.5 Inteligentní senzory teplot	7
3.6 Výběr senzoru	7
4 Hardwarový návrh termostatu.....	8
4.1 Výroba DPS.....	8
4.2 Řídicí jednotka.....	8
4.3 Teplotní čidlo.....	9
4.3.1 ROM příkazy	9
4.3.2 Funkční příkazy	10
4.3.3 Sběrnice one-wire	10
4.4 Jednoduchá klávesnice.....	11
4.4.1 Ošetření zákmitů	11
4.5 LCD displej.....	11
4.6 USB	13
4.7 RTC	14
4.8 Napájení.....	15
5 Návrh softwaru.....	16
5.1 Prostředky pro vývoj.....	16
5.2 Řídicí moduly	16
5.2.1 Hlavní program.....	17
5.2.2 Knihovna DS18B20.....	19
5.2.3 LCD modul	20
5.2.4 RTC modul	21

5.3	Program na PC	22
5.3.1	Knihovna FTD2XX_NET.DLL	24
5.3.2	Knihovna WPF Toolkit.....	24
6	Závěr	25

1 Úvod

Cílem této práce je vyrobit pokojový termostat pro inteligentní regulaci teploty v domě, bytu či v jiných prostorách. Samotné výrobě termostatu však nejprve předchází návrh termostatu, desky plošných spojů a správný výběr součástek. Největší důraz je pak kladen na výběr teplotního senzoru a mikrokontroléru. Poslední neméně potřebnou částí je vytvořit software, který bude termostat pohánět a software na počítači, který bude s termostatem komunikovat.

Textová část je rozdělena do následujících šesti kapitol. V První kapitole nazvané úvod je čtenář stručně uveden do řešené problematiky a je seznámen se zadáním bakalářské práce. Ve druhé kapitole jsou zmíněny některé základní pojmy související s tématem bakalářské práce a jejich vysvětlení. Třetí kapitola je věnována problematice teplotních čidel a senzorů. Jsou vyjmenovány typy teplotních senzorů a čidel, jejich parametry a u každého z nich jsou zmíněny jeho klady a zápory. V závěru kapitoly je pak stěžejní část, která je věnována výběru čidla. Ve čtvrté kapitole je zdokumentován celkový návrh hardwaru termostatu od návrhu desky plošného spoje, přes výběr a popis konkrétních součástek, až po výrobu termostatu. V páté kapitole je popsán softwarový návrh termostatu. Jak program běžící na termostatu uvnitř mikrokontroléru, tak program běžící na počítači, který s termostatem komunikuje. Poslední, šestá kapitola závěr se věnuje celkovému zhodnocení práce, uvedení dosažených výsledků a popisu možných dalších rozšíření, která by mohla být do budoucna doimplementována.

2 Základní pojmy

2.1 Teplota

Teplota je stavová veličina, která určuje 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. Je to jedna z mála veličin, která se nedá měřit přímo, ale prostřednictvím jiných fyzikálních veličin [1].

2.2 Teplotní stupnice

Kelvinova stupnice – Kelvin byl definován zvolením termodynamické teploty trojného bodu (273,16 K), přičemž počátek termodynamické stupnice je dán hodnotou absolutní nuly (0 K).

Celsiova stupnice – je odvozena od Kelvinovy termodynamické stupnice. Počátek Celsiovy stupnice je přesně o 0,01 K nižší než trojný bod vody, což znamená, že stupnice je vůči Kelvinově stupnici posunuta o 273,15 K. Jednotka Celsiovy stupnice [°C].

Fahrenheitova stupnice – Vychází ze dvou základních referenčních bodů. Z bodu mrazu vody 32 °F a bodu varu vody 212 °F. Tyto referenční body jsou od sebe vzdáleny o 180 stupňů, což znamená, že jeden stupeň Fahrenheita odpovídá $\frac{5}{9}$ stupně Celsia resp. Kelvinu [1].

3 Teplotní čidla a senzory

Senzor teploty je funkční prvek tvořící vstupní blok měřicího řetězce. Teplotní čidlo je pak ta část snímače, která převádí teplotu na elektronický signál a ten je dále zpracován elektronikou měřicího přístroje. Právě senzor teploty rozhoduje o přesnosti měřicího zařízení, o měřicím rozsahu, rychlosti měření a dalších důležitých vlastnostech. Je v přímém kontaktu s měřeným okolím. Dle fyzikálního principu se senzory dělí na odporové, termoelektrické, polovodičové s PN přechodem, dilatační, optické a další. V dnešní době jsou na parametry senzorů kladeny velké nároky. Senzory musí být přesné, lineární, citlivé na malé změny teplot a na tyto změny teplot musejí reagovat co nejrychleji [1]. V textu budou dále rozebrány jednotlivé typy senzorů dle fyzikálního principu.

3.1 Kovové odporové senzory

Principem funkce odporových kovových čidel teploty je teplotní závislost odporu kovu. Kov si lze představit jako soubor kladných iontů umístěných v krystalové mřížce. Základním požadavkem je, aby teplotní součinitel odporu kovu byl co největší, stály a teplotní závislost odporu byla lineární.

Pro malý rozsah teplot od 0 °C až do 100 °C lze používat s určitou nejistotou lineární vztah. Pro teplotní součinitel α pak dle [1] platí:

$$\alpha = \frac{R_{100} - R_0}{100R_0},$$

Kde R_0 je odpor čidla při teplotě 0 °C a R_{100} je odpor čidla při teplotě 100 °C. Dalším základním parametrem odporových snímačů teploty je poměr odporů čidla při teplotách 100 °C a 0 °C. Tento poměr je označen písmenem W . Dle [1] platí:

$$W_{100} = \frac{R_{100}}{R_0},$$

U vyšších rozsahů teplot již lineární vztahy nelze použít. Materiály vhodné pro realizaci kovových odporových senzorů:

- platina
- nikl
- měď

Platinové snímače teploty – Platina se vyznačuje svou chemickou netečností, časovou stálostí a vysokou teplotou tání. Z tohoto důvodu jde v průmyslu o nejčastěji využívaná čidla. U tohoto materiálu lze dosáhnout vysokého stupně čistoty a to od 99,9 % až 99,999 %. Dle normy ČSN 751 je poměr elektrického odporu stanoven na $W_{100} = 1,385$, při odporu $R_0 = 100 \Omega$ [1].

Kromě základní hodnoty odporu R_0 se snímače vyrábějí se základní hodnotou 50, 200, 500, 1000 a 2000 Ω . Dle doporučení IEC – 751 se Pt měřicí odpory pro průmyslové použití dělí do dvou tolerančních tříd. Třída A je stanovena pro rozsahy teplot od -200 °C do 650 °C a třída B pro rozsah od -200 °C do 850 °C.

Třída přesnosti A: $\Delta t = \pm(0,150 + 0,002 \cdot |t|)$ (°C),

třída přesnosti B: $\Delta t = \pm(0,300 + 0,005 \cdot |t|)$ (°C).

Niklové snímače teploty – Jsou vyráběny tenkovrstvou technologií. Výhodou těchto snímačů je velká citlivost, rychlá odezva a malé rozměry. Nevýhodou je omezený teplotní rozsah a vůči platině značná nelinearita, horší dlouhodobá stabilita a odolnost vůči působení prostředí.

Niklové snímače jsou dostupné ve stejných R_0 jako platinové snímače. Dovolené tolerance odporu Ni čidla dle normy DIN 43760 jsou:

$$\Delta t = \pm(0,4 + 0,007 \cdot |t|) \text{ } (^{\circ}\text{C}) \text{ pro } > 0 \text{ } ^{\circ}\text{C},$$

$$\Delta t = \pm(0,4 + 0,028 \cdot |t|) \text{ } (^{\circ}\text{C}) \text{ pro } < 0 \text{ } ^{\circ}\text{C}.$$

Měděné snímače teploty – Používají se pro teplotní rozsah od $-200 \text{ } ^{\circ}\text{C}$ do $+200 \text{ } ^{\circ}\text{C}$. V rozmezí teplot od $-50 \text{ } ^{\circ}\text{C}$ do $+150 \text{ } ^{\circ}\text{C}$ vykazuje měděné čidlo lineární průběh.

Nevýhodou těchto čidel je nízká rezistivita (6x menší než u platiny) a vysoká náchylnost k oxidaci mědi. Díky těmto nevýhodám se běžně tato čidla nevyrábějí [1].

3.2 Polovodičové odporové senzory

Polovodičové odporové senzory využívají stejně jako kovové odporové senzory teplotní závislosti odporu. Polovodičové odporové senzory teploty lze rozdělit:

Termistory

- Negastory
- Pozistory
- Monolitické odporové senzory

3.2.1 Termistory

Termistor je tepelně závislý odpor z polovodičových, feroelektrických keramických materiálů. Termistory se vyrábějí ve tvarech disku, destičky, kapky, válečku, aj. malých rozměrů.

Výhody termistoru jsou velká teplotní citlivost, malé rozměry, jednoduchý převod odporu na elektrické napětí nebo proud a možnost přímého měření odporu termistoru na větší vzdálenost. Obrovskou nevýhodou termistorů je velká nelinearita, která omezuje rozsah teploty.

Termistor se dělí na dva typy, dle znaménka teplotního součinitele odporu, na negastor a pozistor.

Negastory (NTC termistory) – Vyrábějí se práškovou technologií z oxidu kovů, např. $\text{Fe}_2\text{O}_3 + \text{TiO}_2, \text{MnO} + \text{CoO}$. Teplotní rozsahy NTC jsou od běžných $-50 \text{ } ^{\circ}\text{C}$ do $+150 \text{ } ^{\circ}\text{C}$. Kromě NTC pro běžné rozsahy se vyrábí i speciální pro nízké (od 4 K) a vysoké teploty (cca do $1000 \text{ } ^{\circ}\text{C}$). Jejich teplotní součinitel odporu je záporný což znamená, že s rostoucí teplotou se zvyšuje koncentrace nosičů náboje a odpor klesá dle vztahu:

$$R_1 = R_2 e^{B\left(\frac{1}{T_1} - \frac{1}{T_2}\right)}$$

kde R_1 je odpor při teplotě T_1 , R_2 je odpor při teplotě T_2 , B je teplotní součinitel.

Při zanedbatelném ohřevu NTC je závislost odporu na teplotě exponenciální, se záporným teplotním součinitelem odporu.

$$R_T = A e^{\frac{B}{T}},$$

kde A je konstanta závislá na geometrickém tvaru a materiálu termistoru, B je teplotní konstanta dána materiálem termistoru a T je termodynamická teplota v kelvinech.

Pozistory (PTC termistory) – jsou termistory s kladným teplotním součinitelem odporu. Vyrábí se z polykrystalické feroelektrické keramiky. Nevýhodou PTC je obrovská nelinearita v závislosti odporu na teplotě. Jejich odpor nejprve s rostoucí teplotou mírně klesá. Od tzv. Curieovy teploty dochází ke strmému nárůstu odporu materiálu [1].

3.2.2 Polovodičové monokrystalické senzory

Monokrystalické senzory teplot lze zhotovit z křemíku, germania, india, včetně jejich slitin.

Monokrystalické Si senzory – V průmyslové praxi se používají k měření teploty v rozsahu od $-50\text{ }^{\circ}\text{C}$ do $+150\text{ }^{\circ}\text{C}$.

Senzory SiC – Používají se pro větší rozsah teplot od $-100\text{ }^{\circ}\text{C}$ do $+450\text{ }^{\circ}\text{C}$. Vyrábějí se tenkovrstvou technologií, vysokofrekvenčním napařováním na substrát Al_2O_3 [1].

3.3 Monolitické PN senzory

Integrované monolitické senzory teploty jsou nejčastěji založeny na teplotní závislosti napětí PN přechodu v propustném směru. Tyto senzory mají teplotní rozsah od $-55\text{ }^{\circ}\text{C}$ do $+150\text{ }^{\circ}\text{C}$. Vyrábějí se ve formě křemíkové diody, tranzistoru a integrovaného obvodu.

3.4 Termoelektrické články

Termoelektrický článek je tvořen dvěma vodiči z různých kovových materiálů, které jsou na obou koncích spolu vodivě spojeny. Jestliže teplota jednoho spoje bude různá od teplot druhého spoje, vzniká termoelektrické napětí a obvodem prochází termoelektrický proud. Materiál na výrobu termoelektrických článků by měl vykazovat velký lineární přírůstek v závislosti na teplotě, stabilitu údaje při dlouhodobém provozu a odolnost proti chemickým a mechanickým vlivům [2].

3.5 Inteligentní senzory teplot

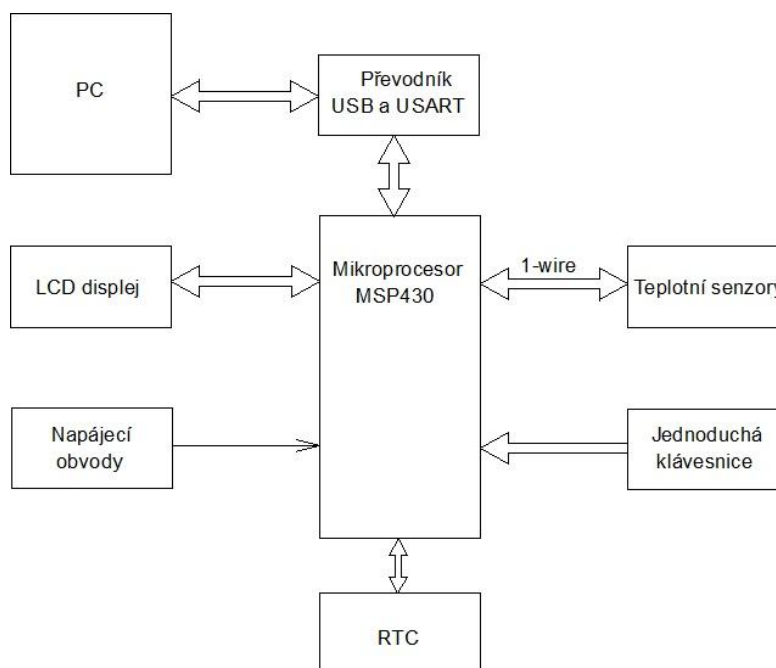
Intelligentní senzory teplot integrují kromě samotného senzoru i jeden nebo několik mikroprocesorů, které se starají o úpravu vstupního signálu (zesílení, linearizace charakteristiky, přizpůsobení rozsahu, aj.), zajišťují i automatickou kalibraci, digitální výstup. Velmi důležitá je obousměrná komunikace mezi inteligentním senzorem a nadřazeným řídicím systémem, která výrazně zvyšuje komfort obsluhy. I přes tyto veškeré poměrně složité zapojení jsou inteligentní senzory velice kompaktní přístroje s malými rozměry [1].

3.6 Výběr senzoru

V této kapitole byly zmíněny některé nejpoužívanější senzory, principy na kterých pracují a jejich výhody a nevýhody. Z výše popsaných byl vybrán jeden z *intelligentních senzorů teploty*, jelikož práce s těmito senzory je jednoduchá, pohodlná a nepřináší s sebou žádné komplikace jako např. linearizaci charakteristiky, či převod analogové hodnoty do digitálního tvaru. Konkrétně bylo vybráno čidlo DS18B20 od firmy Dallas Semiconductor.

4 Hardwarový návrh termostatu

Celé zapojení je složeno z několika funkčních bloků (Obr. 4.1). Řídicím prvkem celého zapojení je 16-bitový mikrokontrolér rodiny MSP430 od Texas Instruments. Jeho úkolem je inicializace obvodu a zajištění komunikace mezi počítačem a ostatními bloky obvodu a čidlem. Dalšími částmi zapojení jsou teplotní čidla posílající do řídicí jednotky údaje o teplotě v okolí, jednoduchá klávesnice, která slouží pro interakci s uživatelem. Dále je to LCD displej který zobrazuje aktuální stav termostatu a další podpůrné obvody. Všechny tyto obvody jsou umístěny na desce plošných spojů.



Obr. 4.1: Blokové schéma termostatu

4.1 Výroba DPS

Jelikož součástí této bakalářské práce bylo vyrobit termostat (elektrický obvod), bylo třeba navrhnout pro něj desku plošných spojů, kam se všechny obvody umístí. Deska byla navržena v programu *EAGLE verze 6.3.0*. Požadavkem bylo navrhnout oboustranný plošný s *rozlitou zemí*, kvůli odstínění navrženého zapojení. Veškeré součástky jsou rozmístěny na jedné straně plošného spoje. Záměrně byly z převážné části voleny součástky v provedení SMD, jen některé jsou v klasickém provedení (např. teplotní senzor, tlačítka, USB konektor). Samotná fyzická výroba desky pak byla provedena technologií vyřezání plošného spoje. Osazení a zapájení součástek na desce plošných spojů probíhalo ručně za použití mikropájky ve školní laboratoři.

4.2 Řídicí jednotka

Nejdůležitější jednotku termostatu tvoří právě řídicí jednotka. Na výběr této součástky byly kladeny největší nároky právě proto, že se stará o správný běh programu a celého zařízení. Pro výběr mikrokontroléru se nabízelo několik možností. Vybíralo se z mikrokontrolérů od firem AVR, FreeScale a Texas Instruments. Jelikož s posledním zmiňovaným mám největší zkušenosti, vybral

jsem mikrokontrolér od této firmy. Konkrétně byl vybrán úsporný mikrokontrolér *MSP430f249* rodiny ultra-low power. Jedná se o 16-bitový mikroprocesor z řady MSP430. Tento mikroprocesor obsahuje interní oscilátor, který může běžet na frekvenci až 16 MHz. Z vnějšku tedy není třeba žádný oscilátor připojovat. Lze však připojit i oscilátor externí. Pro tento účel je MSP430 vybaveno dvěma vstupy. První slouží pro oscilátor, který udává takt pro MCU. Ke druhému se zpravidla připojuje oscilátor s menší frekvencí, který udává frekvenci ACLK, a lze jej použít např. pro chod RTC nezávisle na frekvenci MCU. Mikrokontrolér je vybaven interní programovou pamětí flash o velikosti 60 kB a interní RAM pamětí velikosti 2 kB. Dále MCU obsahuje šest osmibitových I/O portů, což je dostačující počet pro připojení veškerých potřebných komponent [14][13].

4.3 Teplotní čidlo

Důležitým vstupním členem je teplotní čidlo. Teplotní čidlo je elektronická součástka, která převádí fyzikální veličinu (teplo) na elektrickou veličinu. V kapitole 2 jsou popsány jednotlivé druhy teplotních senzorů. Na základě tohoto rozboru jednotlivých senzorů byl vybrán ten nejvhodnější z nich. Byla zvolena varianta digitálního teplotního senzoru. Konkrétně se jedná o programovatelný teploměr DS18B20 od firmy Dallas Semiconductor. Čidlo je dostupné ve třech typech provedení pouzder – 8-pin SO (150mil), 8-pin μ SOP a 3-PIN TO-92. Pro tuto bakalářskou práci byla zvolena třetí varianta.

Čidlo měří teplotu v rozmezí od $-45\text{ }^{\circ}\text{C}$ do $125\text{ }^{\circ}\text{C}$. Pro interval od $-10\text{ }^{\circ}\text{C}$ po $85\text{ }^{\circ}\text{C}$ má čidlo zaručenu minimální přesnost $\pm 0,5\text{ }^{\circ}\text{C}$.

Práce s tímto senzorem je jednoduchá z toho důvodu, že obdržené hodnoty již není třeba nijak speciálně zpracovávat, abychom dostali výslednou teplotu. Senzor totiž odesílá již hodnotu teploty zakódovanou v binární podobě. Komunikace s MCU probíhá po sběrnici one-wire speciálně vyvinutou firmou Dallas Semiconductor.

Komunikace s čidlem probíhá pomocí dvou základních skupin příkazů. Dělí se na *ROM příkazy*, které umožňují získat unikátní 64-bitové adresy jednotlivých teplotních senzorů a na *funkční příkazy*, které řídí činnost teplotního čidla. ROM příkazy masterovi také umožňují zjistit kolik zařízení a jakého typu se na sběrnici nacházejí nebo vyvolalo-li zařízení alarm. ROM příkazů je celkem pět a každý je 8 bitů dlouhý [9].

4.3.1 ROM příkazy

V následujícím textu si představíme všech pět příkazů pracujících s ROM pamětí a popíšeme si, co jednotlivé funkce znamenají a provádějí.

SEARCH ROM [F0h] – Používá se hned po inicializaci komunikace se zařízeními a slouží pro zjištění unikátních 64-bitových ROM kódů každého z připojených zařízení a k určení typu zařízení. Tento příkaz musí být v cyklu proveden tolikrát, kolik je na sběrnici připojeno zařízení. Master si jednotlivé kódy zařízení uloží. Tento příkaz je tedy nutné provádět vždy pouze jednou. Jak z popisu tohoto příkazu vyplývá, používá se jen tehdy, je-li ke sběrnici připojeno více než jedno zařízení. Tento příkaz je poněkud složitější pro implementaci. Detaily implementace lze nalézt na této stránce [8].

READ ROM [33h] – Podobně jako předchozí příkaz umožňuje Masterovi získat 64-bitový ROM kód. Tento příkaz může být použit pouze v případě, je-li na sběrnici připojeno pouze jedno zařízení. Pokud by byl tento příkaz použit na sběrnici s více zařízeními, došlo by ke kolizi dat, jelikož by se všechna zařízení na sběrnici pokoušela komunikovat ve stejnou dobu.

MATCH ROM [55h] – Zápisem tohoto příkazu a následně zápisem unikátního 64-bitového kódu dojde k adresaci konkrétního zařízení na sběrnici. Ostatní zařízení od tohoto okamžiku nebudou

odpovídat, až do doby, dokud na sběrnici nebude proveden reset. Tento příkaz je nutné použít, pokud chceme komunikovat se zařízením po sběrnici, kde je připojeno více než jedno zařízení.

SKIP ROM [CCh] – Tento příkaz lze použít tehdy, když chceme komunikovat se všemi zařízeními na sběrnici současně. Například chceme, aby všechna teplotní čidla začala s převodem teploty nebo pokud je na sběrnici připojeno pouze jedno zařízení. Čtení informace ze zařízení může v tomto případě nastat pouze tehdy, je-li ke sběrnici připojenou pouze jedno zařízení, jinak by došlo ke kolizi dat. V případě, že je na sběrnici připojeno pouze jedno zařízení, šetří tento příkaz čas, jelikož není nutné vysílat 64-bitový unikátní kód.

ALARM SEARCH [ECh] – Tento příkaz je identický s příkazem *SKIP ROM* kromě toho, že na tento příkaz odpoví pouze zařízení, která mají nastavený *alarm flag*. Ten je nastaven, pokud v předchozím měření dojde k přetečení teploty mimo nastavené meze [9].

4.3.2 Funkční příkazy

V této sekci si představíme funkční příkazy, které na rozdíl od ROM příkazů slouží k tomu, aby mohl master zapisovat a číst ze „scratchpad“ paměti teplotního čidla, zahajovat teplotní konverze aj. Tyto příkazy obecně slouží k určování činnosti čidla.

CONVERT [44h] – Po tomto příkazu následuje měření teploty. Hodnota teploty je uložena ve dvou-bytovém registru teploty v „scratchpad“ paměti. Čidlo se poté převede do „low-power“ módu.

WRITE SCRATCHPAD [4Eh] – Tento příkaz masterovi umožňuje zapsat tři byty dat do scratchpad paměti teplotního čidla. První a druhý byte jsou zapsány do registru T_H a T_L , což je rozmezí povolených hodnot teploty, jinak bude spuštěn alarm. Třetí byte slouží pro nastavení režimu konverze.

READ SCRATCHPAD [BEh] – Tento příkaz umožňuje čtení celé scratchpad paměti, tedy všech devět bytů. Čtení začíná od bitu LSB nultého bytu a pokračuje až po devátý byte, který obsahuje CRC kód. Pokud nechceme číst všech devět bytů ale například jen teplotu, lze přenos ukončit vyvoláním resetu na sběrnici.

COPY SCRATCHPAD [48h] – Tento příkaz zkopíruje obsahy registrů T_H , T_L a konfiguračního registru z paměti scratchpad do EEPROM paměti.

RECALL E² [B8h] – Příkaz má opačnou funkci než předešlý zmiňovaný. Tedy obsah registrů T_H , T_L a konfiguračního registru obnoví z paměti EEPROM zpět do scratchpad paměti do jmenovaných registrů.

READ POWER SUPPLY [B4h] – Tento příkaz umožňuje masterovi zjistit, zda se na sběrnici nacházejí nějaká zařízení, která jsou napájena pomocí parazitního režimu. Během čtecího cyklu, zařízení, která jsou napájena pomocí parazitního módu, vysílají logickou nulu [9].

4.3.3 Sběrnice one-wire

Jedná se o sériovou sběrnici, která slouží k připojení různých periférií k mikrokontroléru. Tato sběrnice se používá i u technologie *iButton*, která se používá v elektronických zabezpečovacích systémech (vrátníky, docházkové systémy). Sběrnice se skládá z jednoho řídicího obvodu (master) a jednoho či více ovládaných zařízení (slave). Všechna tato ovládaná zařízení jsou připojena na společnou zem, paralelně zapojena na společný datový vodič a ten je přes „pull-up“ odpor 4,7 k Ω zapojen na napětí, které zdvihá sběrnici do logické 1.

Komunikační protokol je jednoduchý. Komunikaci vždy zahajuje master tím, že stáhne sběrnici do logické 0 a drží ji nejméně 480 μ s. Pak sběrnici uvolní (sběrnice se vrátí zpět na hodnotu log. 1) a naslouchá. Pokud je na sběrnici přítomno nějaké zařízení, detekuje tuto nástupnou hranu a stáhne sběrnici do log. 0 na 60 – 240 μ s. Po tomto „hand shakingu“ master začne vysílat nebo přijímat data.

Data se vysílají v tzv. time slotech, kdy jeden slot trvá 60 až 120 μs a během této doby je přijat nebo odeslán jeden bit informace. Mezi jednotlivými sloty musí být mezera alespoň 1 μs .

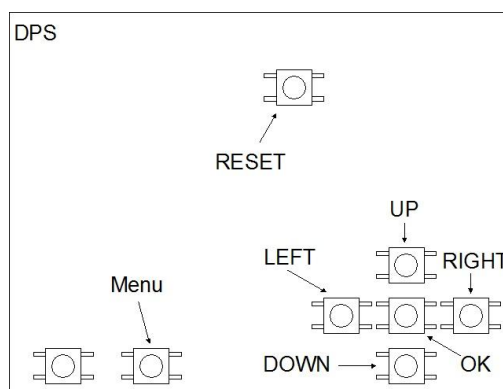
Jelikož lze na tuto sběrnici zapojit více zařízení, je potřeba tyto zařízení od sebe rozeznat. Proto je každé one-wire zařízení vybaveno ROM pamětí která obsahuje 64-bitové unikátní číslo podle kterého se jednotlivá zařízení rozlišují. Pokud je na sběrnici zapojeno pouze jedno zařízení není toto nutné řešit. Jinak se ale komunikace trochu zesložituje tím, že nejprve musíme vyslat příkaz „Match ROM“ a 64-bitový kód zařízení, a až poté můžeme se zařízením komunikovat [3].

4.4 Jednoduchá klávesnice

Jednoduchá klávesnice je realizována pomocí mikropsínačů. Slouží uživateli pro ovládání termostatu, nastavování teploty, hodin. Při stisku tlačítka vzniká parazitní jev, který se nazývá zákmit. Těchto zákmitů při stisku vzniká hned několik (mnoho). Člověk tyto zákmity nemá šanci zaregistrovat, při stisku se mu to jeví jako by ho stiskl jednou, avšak mikroprocesor stihne zaregistrovat veškeré zákmity a jeví se mu to jako by tlačítko bylo stisknuto mnohokrát za sebou. Rozmístění a popis jednotlivých tlačítek na termostatu je vyobrazen na obrázku 4.2.

4.4.1 Ošetření zákmitů

Existují dva způsoby ošetření zákmitů. Prvním z nich je ošetření **softwarově** – po zaregistrování prvního stisku procesorem se procesor zacyklí na nějakou dobu ve smyčce, než zákmity pominou. Nebo **hardwarově** – použitím obvodů, které ošetřují kmitání tlačítek, například připojením kondenzátoru paralelně k tlačítku. V této práci je použit druhý zmiňovaný způsob ošetření zákmitů.

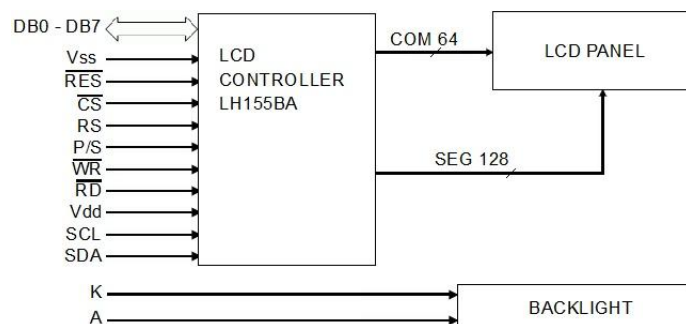


Obr. 4.2: Popis tlačítek na termostatu

4.5 LCD displej

Slouží pro informování uživatele termostatu. Na displej se zobrazují informace o stavu teploty, nastavené teplotě, hodiny, stav kotle a dokonce lze na displeji zobrazit graf průběhu teploty za poslední hodinu běhu termostatu.

Na výběr se nabízelo hned několik typů displejů. Jako nejvhodnější se však kvůli počtu zobrazovaných informací jevil některý z LCD displejů. Právě kvůli poslední zmiňované funkci v předchozím odstavci, bylo třeba zvolit některý z grafických LCD displejů, i když tyto displeje patří do té dražší kategorie. Byl vybrán displej PG12864LRF-NRA-H-Q od firmy Powertip.

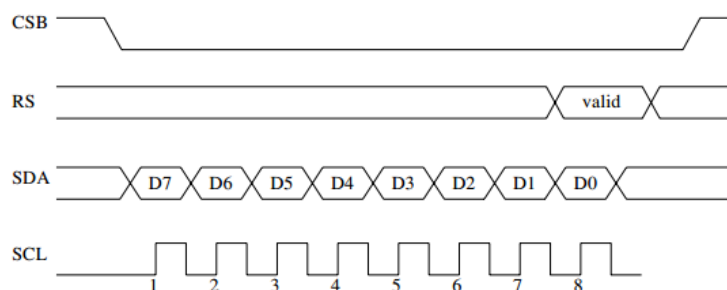


Obr. 4.3: Blokové schéma LCD displeje [4]

Jedná se o transflektní displej s řadičem, konkrétně je to řadič LH155BA od firmy Sharp. Rozměry displeje jsou 128 x 64 bodů, kde první číslo je počet bodů na řádku a druhé znamená počet bodů ve sloupci. Displej je podsvícený zelenými LED diodami. Aby bylo možné na displeji cokoliv vypsát, bylo nutné si nejprve vytvořit vlastní sadu znaků.

U tohoto typu displeje se nabízely hned dvě možnosti komunikace. První z nich je paralelní komunikace. Displej s mikroprocesorem komunikuje buďto pomocí čtyř vodičů, nebo pomocí všech osmi vodičů D0 až D7. Při první zmiňované komunikaci se do displeje zapisuje ve dvou taktech. V prvním taktu jsou zapsány horní čtyři bity a ve druhém taktu spodní 4 bity. Při komunikaci pomocí osmi vodičů jsou do displeje zapsána data během jednoho taktu. Druhým typem komunikace je sériová komunikace. Ke komunikaci jsou využity dva vodiče. Po vodiči SDA jsou posílána data a jsou přečtena při každém taktu na vodiči SCL.

Po zvážení náročnosti plošného spoje byla vybrána sériová komunikace s LCD. Příklad komunikace po sériové lince je vyobrazen na obrázku 4.4.



Obr 4.4: Komunikační protokol LCD po sériové lince [4]

Jak je z obrázku patrné, před samotnou komunikací je třeba nastavit logickou nulu na pinu CS. Nula na tomto pinu je přítomna po celou dobu komunikace. Data jsou posílána po jednom bitu od MSB po LSB a vzorkována na pinu SCL. Pin RS slouží k rozpoznání dat a instrukcí pro displej. Názvy a popis jednotlivých pinů zobrazuje tabulka 4.1.

Vývod	Název	Popis
1	Vss	Napájení 0 V
2	/RES	Reset LCD
3	/CS	Povolení zápisu do displeje
4	RS	Identifikace dat a příkazů pro displej
5	P/S	Přepínání mezi paralelním a sériovým přenosem
6	/WR	Zápis dat
7	/RD	Čtení dat
8 - 15	D0 – D7	Paralelní datová sběrnice
16	Vdd	Napájení +3,3 V
17	SCL	Pin pro vzorkování bitů při sérovém přenosu
18	SDA	Sériový datový vstup
19	A	LED podsvícení displeje (+3,3 V)
20	K	LED podsvícení displeje (0 V)

Tabulka 4.1 Zapojení a popis vývodů LCD displeje [4]

4.6 USB

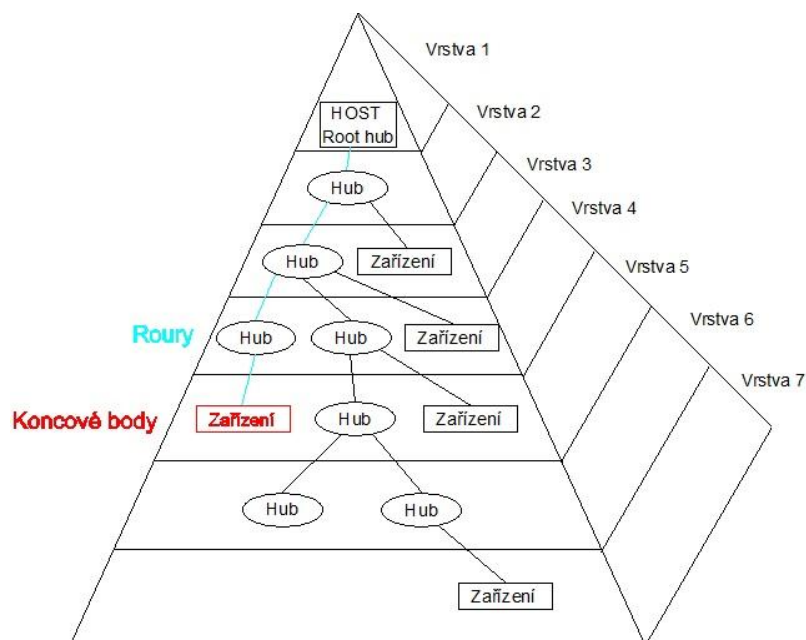
U dnešních zařízení je již USB rozhraní standard. Tato sběrnice se rozmohla díky své obrovské univerzálnosti, což vyplývá i ze samotného názvu sběrnice. Do této sběrnice lze zapojit téměř libovolné zařízení a dokonce lze i zařízení do určitého příkonu přímo z USB napájet. Zařízení může z USB odebírat maximálně proud o velikosti 500 mA. Další obrovskou výhodou této sběrnice je podpora tzv. Plug and Play. Existuje několik typů USB, které jsou popsány v tabulce 4.2. Některé ze zmiňovaných typů již patří spíše mezi archaické. Důležité je, že všechny verze sběrnice jsou zpětně kompatibilní.

Verze	Rychlost	Název
1.0	1,5 Mb/s	Low Speed (LS)
1.1	12 Mb/s	Full Speed (FS)
2.0	480 Mb/s	High Speed (HS)
3.0	4,8 Gb/s	Super Speed

Tabulka 4.2: Verze USB podle rychlosti [5]

V dnešních počítačích je standardem USB 2.0 a v nových počítačích je to pak už USB 3.0, které dosahuje teoreticky rychlosti 10x větší než v případě sběrnice USB 2.0.

Základem USB je zřetěžená hvězdicová topologie. Topologie je vyobrazená na obrázku 4.5. Sběrnice pracuje na principu Master – Slave, kdy slave může komunikovat, jen pokud jej k tomu vyzve master (hostitelský řadič). Ten také vytváří rozhraní mezi počítačem a celou sběrnicí. Přípojným místem USB hostitele je kořenový rozbočovač (Root hub), přes který jde veškerá komunikace. Do USB portu lze zapojit buďto další rozbočovač nebo koncové zařízení (device). Takto lze do USB zapojit až 127 zařízení, kde koncové zařízení se smí nacházet maximálně na sedmé vrstvě, tedy za sebou smí být zapojených maximálně pět zařízení typu rozbočovač, poté už musí být připojeno koncové zařízení.



Obr. 4.5: Topologie USB sběrnice [5]

Rozlišujeme dva typy konektorů. *Type A*, který je vždy připojen na straně směrem ke kořenovému rozbočovači a *Type B*, který je vždy na straně směrem ke koncovému zařízení. Konektory *Type B* pak mohou být ještě ve velikosti *Standard*, *Mini* nebo *Micro*.

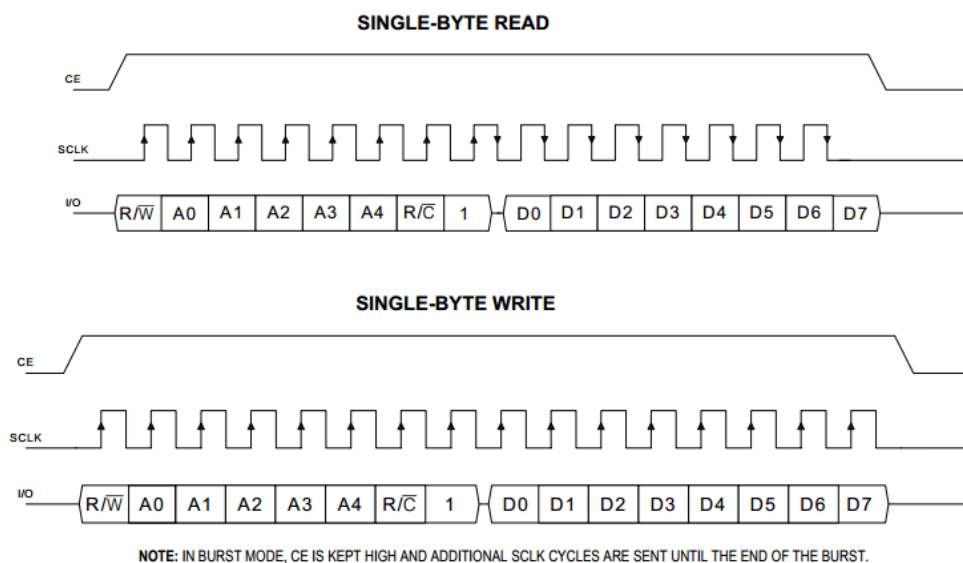
USB rozhraní se skládá ze čtyř vodičů VBus, GND, D+ a D-. Maximální délka vodičů záleží na verzi sběrnice (např. pro USB 2.0 je to max. 5 m). Data se přenášejí diferenciálně po dvou z nich. Logické úrovně těchto vodičů jsou však zcela odlišné od TTL, se kterými pracuje mikroprocesor. Je proto potřeba převést tyto úrovně na hodnoty, kterým bude mikroprocesor rozumět. Toto lze provést hned několika způsoby. Buďto lze zvolit některý z mikroprocesorů, který v sobě obsahuje implementované rozhraní pro komunikaci s USB nebo použít vnější obvod který převádí logické úrovně USB signálů na TTL úrovně. Nejprve bylo zamýšleno použít první způsob, tedy zvolit mikroprocesor s implementovaným rozhraním USB, avšak nabídka těchto mikroprocesorů na trhu je menší než nabídka mikroprocesorů bez implementace tohoto rozhraní a tyto mikroprocesory obsahují zbytečně mnoho integrovaných komponent a jsou zbytečně komplexní. Sice by se celkové zapojení takto zjednodušilo, ale práce s integrovaným USB rozhraním je podstatně složitější než při použití převodníku. Proto bylo v našem případě použito druhé varianty a to převod pomocí externího převodníku FT232RL od firmy FTDI. Použitím tohoto obvodu byl usnadněn i vývoj aplikace na počítači, jelikož počítač vnímá tento čip jako zařízení připojené přes sériovou linku a toto zařízení je schopno fungovat jako samostatné USB zařízení. Obvod pracuje na přenosové rychlosti USB 2.0 (FS). FT232 Je známý a hojně používaný v mnoha různých aplikacích. K mikrokontroléru se připojuje pomocí sériové linky a k jeho provozu je potřeba minimum externích součástek. K obvodu jsou zapojeny i dvě externí LED diody. Jedna indikuje příchozí data a druhá indikuje odchozí data [10].

4.7 RTC

V překladu tato zkratka znamená obvod reálného času. Obvod slouží k uchování aktuálního data a času. Občas bývá tento obvod součástí mikrokontroléru. V našem mikrokontroléru však obsažen není, a proto byl použit externí obvod RTC, konkrétně obvod DS1302 od firmy Dallas Semiconductor. Obvod se vyznačuje tím, že je malý a ke své funkci vyžaduje pouze externě připojený

krystal o frekvenci 32,768 kHz a kapacitě 6 pF. Krystal se připojuje k obvodu na piny X1 a X2. Obvod je vybaven dvěma napájecími piny. Jeden slouží pro připojení primárního napájení a druhý slouží pro připojení záložního napájení. Záložní napájení slouží k tomu, aby při výpadku proudu či vypnutí zařízení nedošlo ke ztrátě informace o aktuálním čase. Je nutné, aby velikost primárního napětí byla alespoň o 0,3 V větší než velikost záložního napětí. Velikost záložního napětí musí být minimálně 2 V. V našem případě je jako záložní zdroj použita baterie CR2430 o velikosti napětí 3 V.

Uvnitř je RTC složeno z registrů ve kterých se uchovávají jednotlivé časové jednotky (vteřiny, minuty, hodiny, ...) a RAM paměti. Uvnitř registrů jsou jednotlivé hodnoty uloženy ve formátu BCD. Obvod reálného času s mikrokontrolérem komunikuje pomocí tří vodičů, /RST, SCLK, I/O. U obvodů novějšího data došlo k přejmenování pinu /RST na CE, funkčnost pinu však zůstala i nadále stejná. K přejmenování došlo pouze proto, aby název lépe charakterizoval funkci pinu. Průběh komunikace je naznačen na obrázku 4.6.



Obr. 4.6: Diagram přenosu dat [6]

Jak lze na obrázku 4.6 vidět, při každém začátku komunikace je nejprve třeba nastavit pin CE (/RST) na log. 1. Poté následuje vyslání adresy registru nebo RAM paměti kam je potřeba přistupovat a následně je provedeno čtení či zápis dat. Každý bit je potvrzován hodinovým signálem na pinu SCLK. Zápis a čtení dat se provádí na pinu I/O [6].

4.8 Napájení

Celé zařízení je postaveno na napájecím napětí o velikosti 3,3 V. Bylo však potřeba i napětí o velikosti 5 V, konkrétně pro běh převodníku FT232RL. Proto byly do návrhu zařazeny dva lineární regulátory napětí. Prvním z nich je obvod L7805 s výstupním napětím 5 V a druhým obvodem je LM1086 s výstupním napětím 3,3 V. obě součástky potřebují ke správné funkci dva tantalové blokovací kondenzátory.

Termostat je možno napájet dvěma způsoby. Prvním způsobem je napájení pomocí síťového napájecího adaptéru. Maximální možné napětí adaptéru, kterým lze zařízení napájet je 12 V, kvůli absenci pasivního chlazení regulátoru. Při tomto druhu napájení je do návrhu zařazena ochranná dioda, která chrání zbytek obvodu proti napěťovým rázům. Druhým způsobem je napájení z USB portu. Výhodou napájení z USB je, že jsou najednou zajištěny dvě funkce. Jednak je celé zařízení napájeno a jednak lze zároveň i komunikovat s počítačem.

5 Návrh softwaru

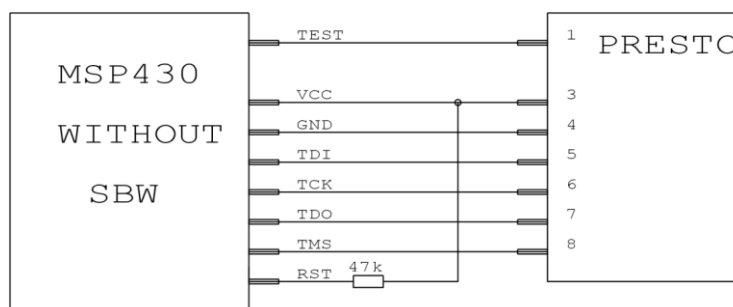
V této části práce budou stručně popsány jednotlivé algoritmy a části softwaru, které souvisí s provozem termostatu. Dále budou popsána vývojová prostředí, ve kterých byly jednotlivé části úlohy programovány, a bude představen programátor, kterým se program do mikrokontroléru nahrával. V poslední řadě bude popsána i počítačová aplikace, která slouží také k ovládání a nastavování termostatu.

5.1 Prostředky pro vývoj

Veškerý software, který běží na vyrobeném termostatu, byl vytvořen v programátorském prostředí Code Composer Studio verze 5.3.0 od Texas Instruments. Tohle prostředí bylo zvoleno právě proto, že je distribuováno stejnou firmou jako programovaný procesor. Vývojové prostředí je firmou nabízeno zdarma buďto na omezenou časovou dobu, nebo na omezenou velikost zkompilevaného kódu.

Nabízelo se hned několik možností v jakém jazyce daný program pro termostat implementovat. Jazyk symbolických instrukcí byl na tuto úlohu příliš složitý, a program v něm by byl značně nepřehledný. Zbývala tedy volba mezi jazyky C a C++. Po delší úvaze se jevila jakožto nejlepší možnost zvolit jazyk C, ve kterém je celý program implementován právě i kvůli zkušenostem s programováním v tomto jazyce. Jelikož jazyk umožňuje rozdělit program do více souborů, stane se program přehledný a čitelný.

Jako programátor, kterým se zkompilevaný program nahraje do procesoru, byl zvolen přípravek PRESTO od firmy ASIX. Tento programátor se může pyšnit obrovskou škálou podporovaných typů procesorů od různých firem. K počítači se připojuje pomocí USB portu, ze kterého je sám napájen. Programování MSP430 probíhá přes rozhraní JTAG. Konkrétní zapojení programátoru k procesoru stejného typu, kterým je vybaven náš termostat, můžeme vidět na obrázku 5.1. Program, kterým se programátor ovládá, má velice jednoduché rozhraní. Program se jmenuje ASIX UP a je zdarma dodáván k programátoru [15].



Obr. 5.1: Zapojení procesoru MSP430, který nemá SBW rozhraní [7]

5.2 Řídící moduly

V této kapitole si popíšeme a blíže nahlédneme do jednotlivých řídicích modulů termostatu. Ať už se bude jednat o části programu, které řídí změření teploty, až po samostatné zobrazení informace na displeji.

5.2.1 Hlavní program

Jelikož se jedná o vestavěný systém, je v mikrokontroléru implementován hlavní program v podobě smyčky, která se neustále provádí dokola. Z této smyčky jsou pak volány ostatní podprogramy nebo přerušení.

Nejprve byl zastaven obvod watchdog, jelikož jeho použití v této aplikaci není potřeba. Nejedná se totiž o nijak kritickou aplikaci, kde by byla potřeba hlídat, zda se nám mikrokontrolér nezasekl. Dále byla nastavena frekvence oscilátoru na 16 MHz, provedeny inicializace jednotlivých komponent jako sériového portu, teplotního senzoru, LCD displeje, čítače a obvodu RTC a byla povolena přerušení od interních komponent [13]. V inicializaci LCD displeje se provede i inicializace bufferu, kde se následně budou ukládat naměřené hodnoty teploty.

Za touto základní inicializací veškerých komponent a základního nastavení následuje ona hlavní smyčka. V této smyčce je nejprve vykreslena základní bitmapa na displej a do ní pak postupně jednotlivé informace. První věc, která je ve smyčce prováděna, je volání funkce pro čtení hodnoty z teplotního čidla a tato hodnota je uložena do proměnné. Upraví se pro zobrazení na displeji a zobrazí se. Provádí se čtení z obvodu reálného času RTC a přečtená hodnota se vykresluje na LCD displej. Jsou taky kontrolovány stavy jednotlivých tlačítek, zda bylo některé tlačítko stisknuto, které tlačítko bylo stisknuto a podle toho se vyvolá požadovaná akce.

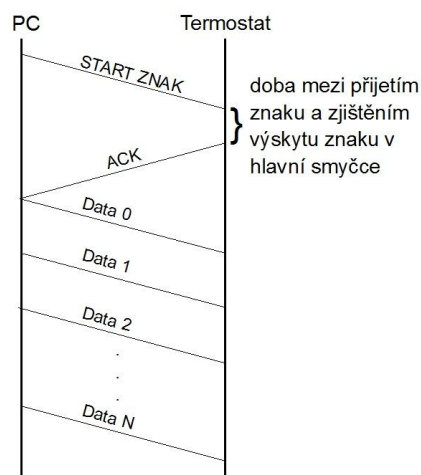
Jelikož jedním z požadavků bylo, aby byl termostat schopen komunikovat s počítačem, jsou v hlavním programu implementovány i prostředky pro inicializaci sériového portu a komunikace po něm. Konkrétně se jedná o odesílání a příjem bytu a o odesílání a příjem řetězce.

Hlavní program také obsahuje několik rutin pro obsluhu přerušení od vnějších zdrojů. První obslužnou rutinou je obsluha přetečení časovače *Timer A*. Přerušení od časovače přichází každou půl vteřinu a právě po uplynutí této doby je rutina vyvolána. V rutině je inkrementována proměnná, kterou odměřujeme 37,5 vteřin a po uplynutí této doby je spuštěna akce pro další měření, neboli rozteč mezi jednotlivými měřeními teplotního čidla je 37,5 vteřiny. Kvůli obhajobě je však tento čas záměrně změněn, aby mohl být průběh ukázán. Původní hodnoty jsou zakomentovány a je nastavena rozteč jednotlivých měření na přibližně jednu vteřinu. Naměřená teplota je uložena do předem nainicializovaného bufferu pro průběh teploty a zobrazení grafu.

Další obslužnou rutinou je rutina pro ukončené vysílání bytu po sériové lince, kdy se kontroluje, zda už je registr, který v sobě obsahuje data pro odeslání, vyprázdněn, tedy všechna data jsou již odeslána, nebo ne.

Posledním podprogramem přerušení je rutina pro příjem znaku na sériové lince. Jelikož data přijímaná z počítače a odesílaná do něj mají určitý komunikační protokol, kontrolujeme první přijaté byty a na jejich základě pak v hlavní smyčce provádíme rozhodnutí co se má stát. Po přijetí bytu nejdříve zkontrolujeme, o kterou operaci se jedná, na základě prvního znaku, a nastavíme příslušnou vlaječku. V hlavním programu tuto nastavenou vlaječku odchytíme a pošleme do počítače paket *Ack*, kterým oznamujeme, že jsme připraveni přijmout data z počítače. Následně počítač pošle svá data. Komunikace mezi počítačem a termostatem je vyobrazena na obrázku 5.2.

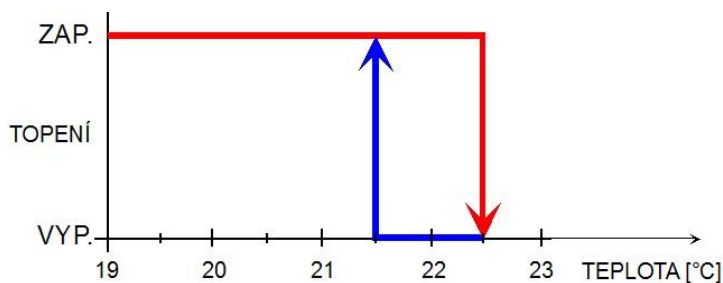
Komunikace termostatu s počítačem



Obr. 5.2: Průběh komunikace mezi počítačem a termostatem

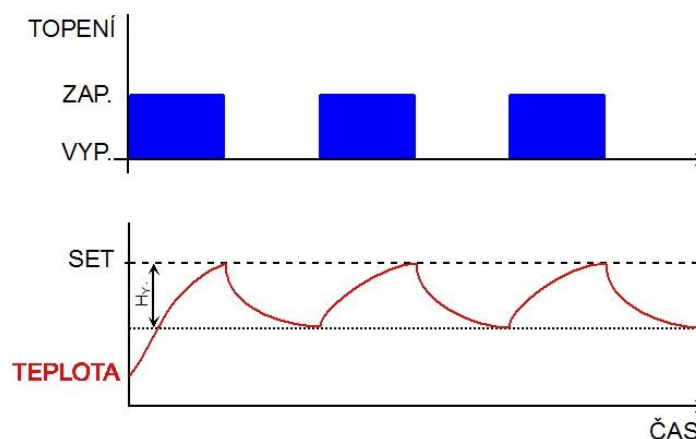
ON/OFF REGULACE – Jedná se o základní typ regulace, zapnuto/vypnuto, též se jí říká dvupolohová regulace. Jestliže poklesne skutečná hodnota teploty pod zadanou hodnotu, přepne se regulátor do stavu ON (zapnuto). Překročí-li skutečná hodnota teploty zadanou veličinu, přepne se regulátor do stavu OFF (vypnuto).

S pojmem ON/OFF regulace se blíže pojí pojem hystereze. Hystereze je vlastně necitlivost na změnu regulované veličiny. Tato vlastnost se do systémů zavádí proto, aby při malých oscilacích nedocházelo k neustálému zapínání a vypínání kotle. Pro topení je typické, že hystereze obvykle leží pod zadanou hodnotou. Jestliže teplota dosáhne nastavené hodnoty, topení se vypne. Pokud teplota po čase klesne o hysterezi, topení se opět zapne. Příklad hystereze je zobrazen na obrázku 5.3, kde je teplota nastavena na 22,5 °C a velikost hystereze je 1 °C.



Obr. 5.3: Reakce akčního členu s hysterezí

Velikost hystereze se volí v závislosti na prostředí, ve kterém je termostat používán a na přesnosti čidla které je uvnitř termostatu zabudováno. Je důležité velikost hystereze dobře zvážit, protože při použití velké hystereze bude docházet k nekvalitní regulaci teploty. Naproti tomu při použití malé nebo žádné hystereze bude docházet k častému spínání kotle a může dojít i k jeho předčasnému zničení. Typický průběh teploty při ON/OFF regulaci je ukázán na obrázku 5.4.



Obr. 5.4: Průběh teploty při ON/OFF regulaci s použitím hystereze

5.2.2 Knihovna DS18B20

Pro komunikaci s teplotním čidlem byla zvolena knihovna, ve které je naimplementována komunikace po sběrnici one-wire. Sběrnice one-wire je popsána výše v kapitole 4.3.3. Knihovna je uložena v souborech *ds18b20.c* a *ds18b20.h*. Obsahuje funkce pro inicializaci, reset sběrnice, zápis a čtení bitu, zápis a čtení bytu, funkci pro převod teploty a pro čtení teploty z čidla [17]. Jednotlivé funkce si dále v textu blíže představíme a popíšeme stručně jejich funkci.

Inicializace sběrnice – funkce, která nastaví pin ke kterému je připojena sběrnice spolu s připojenými zařízeními.

Reset sběrnice – funkce stáhne hodnotu sběrnice na logickou nulu a podrží ji na této hodnotě 480 μ s. Poté se od sběrnice pin odpojí a nechá „pull-up“ odpor aby zvedl hodnotu sběrnice na logickou jedničku.

Zápis bitu – při zápisu bitu jde o to, zda chceme zapsat logickou jedničku nebo nulu. Pokud chceme zapsat logickou jedničku, tak sběrnici stáhneme do logické nuly na 6 μ s. Poté se odpojíme od sběrnice na 64 μ s a necháme ji samovolně přejít do logické jedničky pomocí „pull-up“ odporu. Zápis logické nuly probíhá tak, že převedeme sběrnici do logické nuly a tuto hodnotu na sběrnici podržíme po dobu 60 μ s. Poté se od sběrnice odpojíme na dobu 10 μ s.

Čtení bitu – Při čtení bitu ze slave zařízení opět převedeme sběrnici do stavu logické nuly po dobu 6 μ s, poté se po dobu 9 μ s od sběrnice odpojíme. Po uplynutí této doby přečteme z pinu hodnotu bitu, který teplotní čidlo poslalo. Mezi dvěma čteními bitu je mezera 55 μ s.

Zápis bytu – Probíhá po jednotlivých bitech v cyklu. Z této funkce je tedy volána funkce pro zápis jednoho bitu.

Čtení bytu – Probíhá taktéž po jednotlivých bitech. Z této funkce je v cyklu volána funkce pro čtení jednoho bitu.

Čtení teploty – Tato funkce přečte ze scratchpad paměti čidla první dva byty, ve kterých je uložena hodnota změřené teploty.

Převod teploty – Jelikož je v systému použito pouze jedno teplotní čidlo, je použit příkaz *SKIP ROM* pro adresaci čidla a následně je použit příkaz *CONVERT* pro započítání měření. Měření trvá čidlu zhruba 750 μ s. Po tuto dobu mikroprocesor čeká ve smyčce. Poté spustí reset sběrnice. Opět na sběrnici zapíše příkaz *SKIP ROM* a následně *READ SCRATCHPAD* pro přečtení obsahu registrů. Zavolá se funkce pro přečtení hodnoty teploty a MCU zapíše reset na sběrnici, aby přerušil přenos ostatního obsahu registrů, který nás zde nezajímá. Na přijatou teplotu se aplikuje korekce a hodnota teploty je vrácena.

Jelikož sběrnice pracuje na principu otevřeného kolektoru, znamená to, že hodnota logická nula je nastavena pomocí otevření tranzistoru a uzemnění sběrnice a logická jednička je nastavena pomocí

„pull-up“ odporu, je odpojení od sběrnice realizováno změnou funkce pinu na vstupní. To na sběrnici zajistí hodnotu logické jedničky. Logická nula je pak nastavována převedením pinu na výstupní.

V následujícím kódu je ukázáno přepínání pinu ze vstupu na výstup na bráně P3. Čidlo je připojeno k pinu č. 3 a je pojmenován jako OWPORTPIN.

```
#define OW_LO { // zapis log. 0
    OWPORTDIR |= OWPORTPIN;    // nastaví port na vystup
    OWPORTREN &= ~OWPORTPIN;
    OWPORTOUT &= ~OWPORTPIN; }

#define OW_RLS { // zapis log. 1
    OWPORTDIR &= ~OWPORTPIN;    // nastaví port na vstup
    OWPORTREN |= OWPORTPIN;
    OWPORTOUT |= OWPORTPIN; }
```

V registru *P3DIR*, který je v programu pojmenovaný jako *OWPORTDIR*, se nastavuje směr pinů. Pokud je hodnota určitého bitu vynulována, je pin, namapovaný na tento bit, přepnut do vstupního režimu, pokud je bit nastaven na logickou jedničku, je pin ve výstupním režimu [13].

5.2.3 LCD modul

Obsahuje funkce pro správnou funkci LCD displeje a je implementován v modulech *LCD.c* a *LCD.h*. Jedná se o grafický displej a v knihovně jsou implementovány i funkce pro vykreslení grafu.

Před prvním použitím displeje je nutné displej nejprve inicializovat. Ve funkci pro inicializaci se tedy nejprve ustaví komunikace s displejem, nastaví se příslušné vstupní a výstupní piny. Po správném nastavení pinů na příslušné ovládací piny LCD je potřeba do řadiče LCD zapsat několik řídicích bytů. Bylo nutné nastavit X-ovou a Y-ovou adresu od které se má začít zobrazovat. Dále byly nastaveny *power control registry*.

V následujícím textu si blíže představíme funkce použité v tomto modulu a popíšeme si stručně jejich fungování.

Zápis příkazu/dat – Abychom mohli cokoliv do LCD displeje zapsat, je nejprve nutné převést pin *chip select* na aktivní úroveň. Pin *chip select* je negovaný, tedy na hodnotu logické nuly. Prvním parametrem funkce je proměnná s informací, zda se do displeje má zapsat příkaz nebo data. Podle toho, co proměnná obsahuje je pin RS nastaven buďto do logické nuly, tedy zapisujeme data, nebo do logické jedničky a v tom případě zapisujeme příkaz. Po nastavení těchto řídicích bitů jsou odeslána samotná data. Jelikož je displej s MCU komunikuje po sériové sběrnici, jsou data odesílána po jednom bitu v cyklu od nejvýznamnějšího bitu po nejméně významný. Každý bit je vzorkován hodinovým pulzem.

Zápis pozice – Funkce nastavuje souřadnice na LCD displeji, kam se následně zapisují příchozí data. Má dva parametry, jedním je souřadnice na ose X druhým je souřadnice na ose Y.

Smazání displeje – Funkce smaže veškerý obsah vypsáný na displeji. V cyklu je volána funkce pro zápis dat, a zapisuje se nulový byte.

Výpis znaku – Jednotlivé znaky jsou definovány v modulu *fonts.h*. Funkce má jeden parametr, kterým je znak který chceme vypsát. Tento znak je po úpravě použit k adresování v poli znaků a je vybrána pozice tohoto znaku. V cyklu je poté znak vykreslen.

LCD displej je rozdělen na segmenty. Při použití grafického módu, který je použit i v této práci, je paměť pro grafický displej přístupná po osmi bitech najednou. Rozsah adres osy X je v rozmezí od 00h po 0Fh. Rozsah adres osy Y je od 00h po 3Fh. V modulu *fonts.h* jsou tedy takto uloženy jednotlivé byty (řádky), ze kterých je následně v cyklu výsledný znak poskládán.

Výpis řetězce – Funkce běží v cyklu, ze kterého volá funkci pro výpis jednotlivých znaků z řetězce.

Převod čísla na znaky – Funkce se používá pro převod naměřené teploty na znaky, aby bylo možné teplotu zobrazit. Parametry funkce jsou teplota, počet desetinných čísel a souřadnice, kde má být teplota zobrazena. V cyklu je prováděna operace modulo nad zadanou teplotou a k výslednému číslu je přičítán znak nula. Výsledné znaky jsou ukládány do pole, jehož obsah se nakonec reverzuje a vypíše na displej.

Vykreslení bitmapy – Je funkce, která vykresluje pozadí za textem. Veškeré bitmapy jsou uloženy v modulu *bitmap.h*. Funkce má pouze jeden parametr a tím je číslo bitmapy, která má být vykreslena.

Vykreslení grafu – Nejedná se pouze o jednu funkci nýbrž o soubor funkcí, které se starají o veškeré úkoly, které jsou potřeba pro finální vykreslení grafu. Nejprve je to inicializační funkce, která nastaví jednotlivé prvky v bufferu do výchozí hodnoty. Poté je to funkce pro inicializace samotného grafu. Graf je uložen v poli o velikosti zobrazované plochy, na kterou se vykresluje graf. Jedná se tedy o dvou rozměrné pole.

Před samotným vykreslováním grafu je nejprve potřeba vykreslit osy. Na ose X je vynesena čas, poslední hodina měření. Na ose Y je vynesena velikost teploty v daný čas.

Další funkcí je funkce pro přidání teploty do bufferu. Je volána každých 37,5 vteřin. Celý zaplněný buffer představuje hodnoty teplot za poslední hodinu měření. Uvnitř funkce se volá funkce pro získání teploty z čidla. Jelikož se předpokládá, že termostat bude v provozu déle než jen jednu hodinu, bylo potřeba zajistit odsouvání grafu na displeji, aby vždy byly zobrazeny aktuální hodnoty za poslední hodinu měření. To se provádí v cyklu nad celým polem, ve kterém je graf uložen. V každé položce v poli je uložen jeden byte. Při posouvání je potřeba celé pole posunout o jeden bit doleva. Nad prvním bytem je realizováno jednoduché posunutí. U druhého bytu se nejprve zjistí, zda nemá v bitu, který je nejbližší předchozímu bytu, logickou jedničku. Pokud ano, je tato jednička přičtena předchozímu bytu a tento byte je rovněž posunut o jeden bit, atd. Aby byl graf spojitý a nevypadal pouze jako sled bodů, dopočítávají se a vykreslují se body mezi jednotlivými naměřenými teplotami.

Funkce pro vytváření grafu běží v cyklu, dokud není stisknuto tlačítko OK. V tomto cyklu je vložen další cyklus, který běží, dokud jsou v bufferu hodnoty aktuální hodnoty. Dokud ještě není graf přes celou plochu displeje, dochází k vykreslování nových bodů. Jakmile však dojde k tomu, že graf zaplní celou plochu, je nutné graf odsunout. Po odsunutí grafu je do bufferu uložena nová naměřená hodnota a jsou dopočítány body mezi předposlední hodnotou a poslední naměřenou hodnotou tak, aby to při vykreslení vypadalo, že jde o spojitý průběh. Celé pole je následně vykresleno na LCD displej [16].

5.2.4 RTC modul

Obsahuje funkce pro obsluhu externího RTC obvodu. Funkce jsou implementovány v souborech *ds1302.c* a v hlavičkovém souboru *ds1302.h*. Jedná se o externí obvod, který s mikrokontrolérem komunikuje po sériové sběrnici pomocí tří vodičů.

Dříve než se s obvodem bude pracovat, je nutné jej inicializovat. Inicializací se rozumí ustavení komunikace s obvodem reálného času, nastavení a pojmenování pinů mikrokontroléru na piny RTC. Po nastavení pinů je potřeba v obvodu vynulovat dvojici bitů. Nejprve bit *Write Protection*, aby bylo vůbec možné zapsat do RTC cokoliv jiného. Druhý bit, který je třeba vynulovat, se jmenuje *Clock Halt*. Po vynulování tohoto pinu se nastartuje krystal připojený k RTC a hodiny se rozběhnou. Nakonec se do hodin zapíše aktuální čas a poté už se jen v nekonečné smyčce čte. Před jakýmkoliv

čtením či zápisem do obvodu RTC, je potřeba povolit do něj zápis aktivováním pinu CE, převedením do logické jedničky.

Dále si v textu rozebereme a podrobněji popíšeme jednotlivé funkce použité v tomto modulu.

Zpoždovací smyčka – Funkce má jeden parametr, kterým je čas v mikrosekundách. Tato funkce se používá například pro určení délky vzorkovacího pulzu SCLK.

Zákaz práce s obvodem – Funkci můžeme provést, pokud chceme zakázat práci s obvodem. Hodí se například, pokud máme ke stejným pinům připojeno více zařízení, tak aby zařízení nebylo ovlivňováno zápisem do jiného. To se provádí zápisem logické nuly na pin CE.

Zápis bytu – Jelikož je RTC obvod k mikrokontroléru připojen pomocí sériové sběrnice, přenáší se veškerá data po jednom bitu. Funkce pro zápis bytu má jeden parametr, byte, který chceme zapsat. Před odesláním je potřeba pin SDA přepnout do výstupního režimu. Odesílání je realizováno v cyklu, kdy přenos probíhá po jednom bitu a každý bit je vzorkován na pinu SCLK.

Čtení bytu – Taktéž čtení probíhá ve smyčce a jednotlivé bity jsou potvrzovány hodinovými pulzy na pinu SCLK. Před samotným čtením je potřeba změnit funkci pinu SDA na vstupní.

Zápis dat do RTC – Funkce obsahuje dva parametry. Prvním z nich je adresa, druhým jsou data. Před zápisem dat je resetována komunikace mezi mikrokontrolérem a RTC obvodem. Po ustavení komunikace je do RTC zapsán první byte, který představuje adresu registru, kam se bude zapisovat. Vzápětí jsou vyslána samotná data.

Čtení dat z RTC – Funkce obsahuje jediný parametr, kterým je adresa registru ze kterého se má číst. Před zápisem dat je resetována komunikace mezi mikrokontrolérem a RTC obvodem. Po ustavení komunikace je do RTC zapsána adresa registru, ze kterého se mají číst data, předaná v parametru. Následně je zavolána funkce pro čtení bytu.

Zápis dat do RTC v burst módu – Tento typ RTC obvodu nabízí zápis v tzv. burst módu, což znamená, že není potřeba vystavovat adresy každého registru zvlášť a zvlášť do nich zapisovat data, ale po zapsání bytu *BEh* lze postupně v cyklu zapsat data do jednotlivých registrů bez předchozího zápisu adresy registru.

Čtení dat z RTC v burst módu – Podobně jako funkci zápis v burst módu nám obvod nabízí i čtení v tomto režimu. Do RTC obvodu stačí pouze zapsat byte *BFh*, a pak už v cyklu číst jednotlivé příchozí byty. Byty mají daný přesný sled jak pro čtení, tak pro zápis. Pro obě dvě operace je sled stejný. Přenos začíná nulovým bitem registru vteřin, poté minuty, hodiny, dny, měsíce, dny v týdnu a nakonec roky.

Čtení/zápis dat do RAM v burst módu – Již podle názvu funguje obdobně jako dvě předchozí funkce. Zapsáním bytu *FFh* či *FEh* dojde k zápisu či přečtení všech jedenácti RAM paměťových buněk [18] [9].

5.3 Program na PC

Posledním bodem bakalářské práce bylo navrhnout počítačový software, který by dokázal s termostatem komunikovat, nastavovat jisté parametry a číst je.

Program pro počítač byl vytvořen v jazyce C# s použitím WPF. Jelikož jsou veškeré operace ohledně měření a výpočtů implementovány v mikrokontroléru, je počítačový program velmi jednoduchý. Veškerou komunikaci s čipem FT232RL obstarává knihovna *FTD2XX_NET.DLL*, dodávaná přímo k tomuto obvodu [11]. Ukázka hlavního okna programu je na obrázku 5.5.

Jakákoliv akce je iniciována stiskem tlačítka ve formuláři v programu na počítači. Termostat se pak jen ve smyčce dotazuje, zda neobdržel nějaký z řídicích znaků z počítače. Při zapnutí programu je nejprve zařízení inicializováno a jsou provedena důležitá nastavení pro komunikaci s ním. Program poté čeká na stisk tlačítka, na němž záleží další provedená akce. Aby bylo možné s termostatem komunikovat, musí být termostat přepnutý vždy na základní obrazovce.

Při stisku tlačítka „Aktuální stav“ se do termostatu запиše řídicí znak *r*. Ve smyčce se čeká na počet příchozích bytů. Po přijetí této informace jsou data z termostatu přečtena do počítače. Přečtená data se skládají z aktuální teploty, informace, zda je zapnut kotel, velikosti hystereze a aktuální nastavené teploty. Tato data je nutné analyzovat a rozdělit. Všechny tyto hodnoty jsou od sebe odděleny mezerou, takže za pomoci funkce *Split()*, kterou nám *C#* nabízí, hodnoty rozdělíme do proměnných a jejich obsahy zobrazíme.

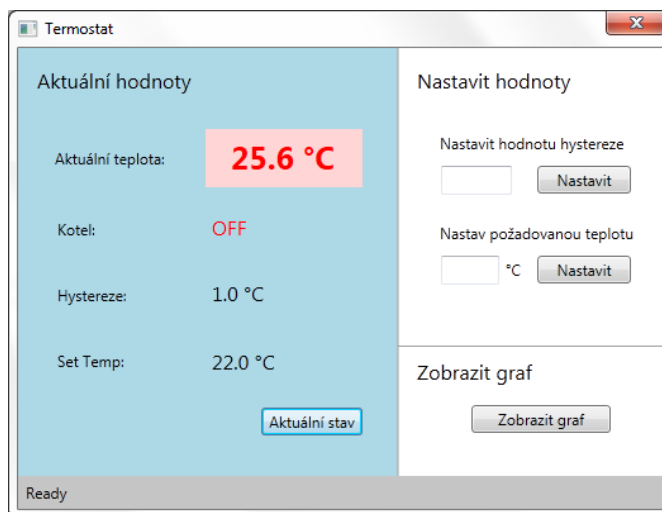
Program nám umožňuje měnit velikost hystereze, která je defaultně nastavena na hodnotu jednoho stupně celsia a teplotu, která je defaultně nastavena na dvaadvacet stupňů celsia.

Při zadání nové hodnoty hystereze a stisknutí tlačítka „Nastavit“, je do mikrokontroléru odeslán řídicí byte *h*. Poté, co tento řídicí byte mikrokontrolér zaregistruje v hlavní smyčce, pošle počítači nazpět paket *Ack*, kterým oznamuje, že je připraven pro přijetí nové hodnoty hystereze. Program na počítači následně odešle novou hodnotu hystereze.

Pokud zadáme novou hodnotu teploty a odešleme jí tlačítkem „Nastavit“, je do termostatu odeslán řídicí byte *s*. Při průchodu hlavní smyčkou v mikrokontroléru je tento řídicí byte zaregistrován, a jako reakce je odeslán paket *Ack*, kterým mikrokontrolér oznamuje, že je připraven přijmout novou hodnotu teploty. Program v počítači následně odešle zadanou hodnotu teploty uživatelem.

Poslední funkcí, kterou daný program disponuje, je zobrazení grafu průběhu teploty za poslední hodinu provozu termostatu. V pravém dolním rohu programu je k tomu určeno tlačítko „Zobrazit graf“. Po stisku tohoto tlačítka je do mikrokontroléru odeslán řídicí znak *g*. V hlavní smyčce je tento byte zaregistrován a mikrokontrolér na něj odpoví odesláním obsahu bufferu s naměřenými teplotami. Tyto hodnoty jsou v programu uloženy do pole a předány druhému oknu, ve kterém bude graf zobrazen. Toto okno obdrží referenci na hlavní okno a je zobrazeno přes celou obrazovku. Předané hodnoty jsou nejprve převedeny z formátu *string* na formát *double*. S těchto hodnot je za použití knihovny *WPFToolkit.dll* zobrazen graf průběhu teploty [12].

Komunikace mezi mikrokontrolérem a termostatem je navržena tak, že veškerý přenos je řízen z počítače. Nejprve počítač odešle identifikační byte, který je v hlavním programu uvnitř mikrokontroléru zachycen a rozezná se, jedná-li se o nastavení teploty, požadavek na velikost teploty, požadavek na odeslání velikosti hystereze, nastavení hystereze, atd. Jedná-li se o operaci zápisu do mikrokontroléru, pošle mikrokontrolér paket *Ack*, kterým říká, že je připraven přijmout příchozí data. V případě, že se jedná o čtení z mikrokontroléru, žádný paket *Ack* se neposílá, ale mikrokontrolér rovnou odesílá požadovaná data. Celý komunikační protokol mezi mikrokontrolérem a počítačem je shrnut v tabulce 5.1.



Obr. 5.5: Hlavní okno programu

Počítač -> mikrokontrolér	Mikrokontrolér -> počítač	Popis
Odeslán byte „r“	Odeslána posloupnost bytu „temp, kotel, hystereze, set temp“	Načtení aktuálního stavu termostatu
Odeslán byte „h“, čekání na paket „Ack“ a po obdržení zápis hystereze	Odeslán paket „Ack“	Zápis nové hodnoty hystereze do termostatu
Odeslán byte „s“, čekání na paket „Ack“ a po obdržení zápis teploty	Odeslán paket „Ack“	Zápis nové hodnoty teploty do termostatu
Odeslán byte „g“	Odeslán obsah bufferu s naměřenými teplotami	Čtení naměřených hodnot z termostatu

Tabulka 5.1: Komunikační protokol mezi počítačem a mikrokontrolérem

5.3.1 Knihovna FTD2XX_NET.DLL

Knihovna vyvinutá firmou FTDI. Je určena pro jazyk C#. Jsou v ní obsaženy veškeré funkce potřebné pro komunikaci s čipem FT232RL přes USB sběrnici. Ke knihovně je dostupná dokumentace ve formě XML souboru [11].

5.3.2 Knihovna WPF Toolkit

WPF Toolkit je sada funkcí a WPF komponent od Microsoftu, které jsou k dispozici mimo .NET Framework. Tato knihovna poskytuje spoustu funkcí a modulů, jako např. již zmiňovaný a námi používaný modul pro tvorbu grafů.

Předtím než je možné použít jakoukoli funkci z modulu pro vykreslení grafu, je nutné stáhnout si tuto knihovnu a v programu přidat referenci na potřebný modul. Z této knihovny jsou v našem programu použity prostředky pro tvorbu grafů [12].

6 Závěr

Cílem této práce bylo vytvořit funkční pokojový termostat. Na začátku práce je nejprve popsána teorie měření teploty a výběr nejvhodnějšího teplotního čidla, jemuž byla přikládána velká pozornost. V práci byly podrobně popsány různé druhy teplotních senzorů a čidel. Po zvážení všech kladů a záporů různých typů senzorů a čidel byl vybrán inteligentní senzor DS18B20.

Další kapitola byla věnována hardwarovému návrhu. Výsledkem této části bylo elektronické schéma, na jehož základě došlo k výrobě fyzické podoby termostatu. Jako řídicí jednotka byl vybrán 16-bitový mikrokontrolér MSP430F249 od firmy Texas Instruments. Termostat byl dále vybaven grafickým LCD displejem o rozměrech 128x64 pixelů pro zobrazování aktuálního stavu uživateli, jednoduchou klávesnicí pro ovládání termostatu, obvodem reálného času a rozhraním pro komunikaci po sběrnici USB s počítačem. Z vybraných komponent byl sestaven prototyp termostatu.

V další kapitole byl pro oživení termostatu navrhován software. Díky dobrému návrhu softwaru je termostat schopen měřit teplotu, přijímat data nastavená uživatelem na jednoduché klávesnici a je schopen komunikace s počítačem. Navíc je termostat schopen zobrazit na LCD displeji graf průběhu teploty za poslední hodinu provozu. Pro regulaci teploty byl navržen ON/OFF algoritmus s přidáním záporné hystereze. Tento algoritmus, oproti klasickému ON/OFF regulátoru, eliminuje vliv setrvačnosti soustavy a zajišťuje kolísání teploty přesně podle přednastavené hystereze. Pro počítač byla vyvinuta aplikace, která umožňuje číst nastavené parametry z termostatu a měnit je. Také dokáže načíst hodnoty teploty naměřené za poslední hodinu provozu a zobrazit je jako graf.

Po oživení termostatu se ukázalo pár skrytých problémů. Na termostatu se nepodařilo zprovoznit RTC obvod DS1302. Zápis do tohoto obvodu a jeho čtení fungovalo bezproblémově, avšak nikdy nedošlo ke spuštění a rozkmitání krystalu. Nejprve došlo k výměně krystalu, to však bohužel nevedlo k žádné změně. Nakonec byl vyměněn celý obvod RTC za nový, avšak ani toto nepomohlo a přetrvával stále stejný problém. Po delším zkoumání na internetu bylo zjištěno, že daný obvod je velmi citlivý na výběr krystalu, hlavně na jeho kapacitu. Krystal je potřeba připojovat v bezprostřední blízkosti vývodů a musí mít kapacitu 6 pF. Nefunkčnost obvodu je tedy přisuzována špatné volbě krystalu s příliš vysokou kapacitou a nevhodnou polohou na DPS. Dalším problémem byla obrovská nepřesnost teplotního senzoru. I když je v manuálovém listu uvedeno, že jde o velmi přesnou součástku s nepřesností maximálně $\pm 0,5$ °C, tak se teplota naměřená tímto čidlem lišila od pokojové teploty přibližně o 16 °C. V tomto případě bohužel nebyla zjištěna příčina tohoto markantního rozdílu, a tak nezbyvalo nic jiného, než od naměřené hodnoty odečíst konstantu právě o velikosti 16 °C. Při častém měření teploty na senzoru navíc docházelo k postupnému ohřevu součástky a tedy k ještě větší nepřesnosti měření. Tato špatná vlastnost byla eliminována méně častým prováděním měření na teplotním čidlu. Nejprve k měření docházelo zhruba každou vteřinu a senzor se tak velmi rychle zahřál. Doporučená doba mezi jednotlivými měřeními, aby nedocházelo k ohřevu senzoru, je deset vteřin. V našem případě senzor měří teplotu každých 37,5 vteřin, což je dostatečně dlouhá doba a tento parazitní jev byl tímto zcela eliminován.

Termostat byl testován v reálných podmínkách a ukázal se jako funkční. Jediné co je potřeba k termostatu přidat je výstup na spínací člen, zatím je vyveden jen ve formě LED diody, aby demonstroval spínání a rozepínání silového obvodu. Po zprovoznění RTC obvodu by bylo možné využívat přípravek jako plnohodnotný termostat i s nastavováním denního a nočního režimu.

Jelikož jde pouze o řídicí část termostatu, jeví se jako adekvátní rozšíření výroba silové, spínací části termostatu. Dalším rozšířením by mohlo být zapojení více čidel DS18B20, jelikož je lze připojit všechny na stejnou sběrnici a vyrobit tak inteligentní snímání teploty např. v každé místnosti a popřípadě i venkovní teploty. Posledním rozšířením by byla implementace rozhraní Ethernet a ovládání termostatu přes web odkudkoliv na světě.

Literatura

- [1] KREIDL, Marcel. Měření teploty - senzory a měřicí obvody. 1. vyd. Praha : BEN, 2005. 240 s. ISBN 80-7300-145-4.
- [2] KADLEC, Karel, MILOŠ, Kmínek. Měřicí a řídicí technika. *Učebnice* [online]. 2005[cit. 2013-02-17]. Dostupné z: <http://uprt.vscht.cz/ucebnice/MRT/>
- [3] MALÝ, Martin. Sběrnice 1-Wire™. *HW.cz | Vše o elektronice a programování* [online]. 2004 [cit. 2013-04-22]. Dostupné z: <http://www.hw.cz/navrh-obvodu/rozhrani/sbornice-1-wiretm.html>
- [4] Datasheet PG12864LRF-NRA-H-Q. *Farnell* [online]. 2006 [cit. 2013-04-23]. Dostupné z: <http://www.farnell.com/datasheets/1310996.pdf>
- [5] STRAKA, Martin, Václav ŠIMEK a Jan KAŠTIL. Analýza komunikace na sběrnici USB. *Fakulta informačních technologií VUT v Brně* [online]. 2008 [cit. 2013-04-24]. Dostupné z: <http://www.fit.vutbr.cz/study/courses/IPZ/public/laboratore/rok2008/ipz-usb-prez.pdf>
- [6] DS1302 Trickle-Charge Timekeeping Chip. *Analog, linear, and mixed-signal devices from Maxim* [online]. 2008 [cit. 2013-04-24]. Dostupné z: <http://datasheets.maximintegrated.com/en/ds/DS1302.pdf>
- [7] Programátory ASIX: Referenční příručka. *ASIX* [online]. 1991, 2012 [cit. 2013-04-26]. Dostupné z: http://www.asix.cz/download/programmers/programmers_cz.pdf
- [8] Book of iButton® Standards. *Analog, linear, and mixed-signal devices from Maxim* [online]. 2002 [cit. 2013-04-28]. Dostupné z: www.maxim-ic.com/ibuttonbook
- [9] DS18B20 Programmable Resolution 1-Wire Digital Thermometer. *Analog, linear, and mixed-signal devices from Maxim* [online]. 2008 [cit. 2013-04-24]. Dostupné z: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [10] FT232R USB UART IC. *Future Technology Devices International Ltd.* [online]. 2010 [cit. 2013-05-05]. Dostupné z: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf
- [11] Code Examples. *Future Technology Devices International Ltd.* [online]. 2010 [cit. 2013-05-05]. Dostupné z: <http://www.ftdichip.com/Support/SoftwareExamples/CodeExamples/CSharp.htm>
- [12] Windows Presentation Foundation - Download WPF Toolkit - June 2009 Release. *CodePlex - Open Source Project Hosting* [online]. 2009 [cit. 2013-05-07]. Dostupné z: <http://wpf.codeplex.com/releases/view/29117>
- [13] MSP430x2xx Family User's Guide. *Analog, Embedded Processing, Semiconductor Company, Texas Instruments - TI.com* [online]. 2004, 2012 [cit. 2013-05-07]. Dostupné z: <http://www.ti.com/lit/ug/slau144i/slau144i.pdf>
- [14] MSP430F23x MSP430F24x(1) MSP430F2410 MIXED SIGNAL MICROCONTROLLER. *Analog, Embedded Processing, Semiconductor Company, Texas Instruments - TI.com* [online]. 2007, 2012 [cit. 2013-05-07]. Dostupné z: <http://www.ti.com/lit/ds/symlink/msp430f249.pdf>
- [15] ASIX: PRESTO. *ASIX* [online]. 2013 [cit. 2013-05-07]. Dostupné z: http://www.asix.cz/prg_presto.htm
- [16] LH155BA. *ELECTRONIC ASSEMBLY : making things easier! Grafik Display, Serielle Grafikdisplays LCD Module, Dotmatrix, Grafik Module, Digitalvoltmeter, Ereigniszähler* [online]. 2006 [cit. 2013-05-07]. Dostupné z: <http://www.lcd-module.de/eng/pdf/zubehoer/lh155.pdf>

- [17] 1-Wire and MSP430. *Small Bulb* [online]. 2012 [cit. 2013-05-07]. Dostupné z: <http://www.smallbulb.net/2012/238-1-wire-and-msp430>
- [18] MSP430 DS1302. *Pudn* [online]. 2009 [cit. 2013-05-07]. Dostupné z: <http://www.pudn.com/downloads187/sourcecode/embed/detail877316.html>

Seznam příloh

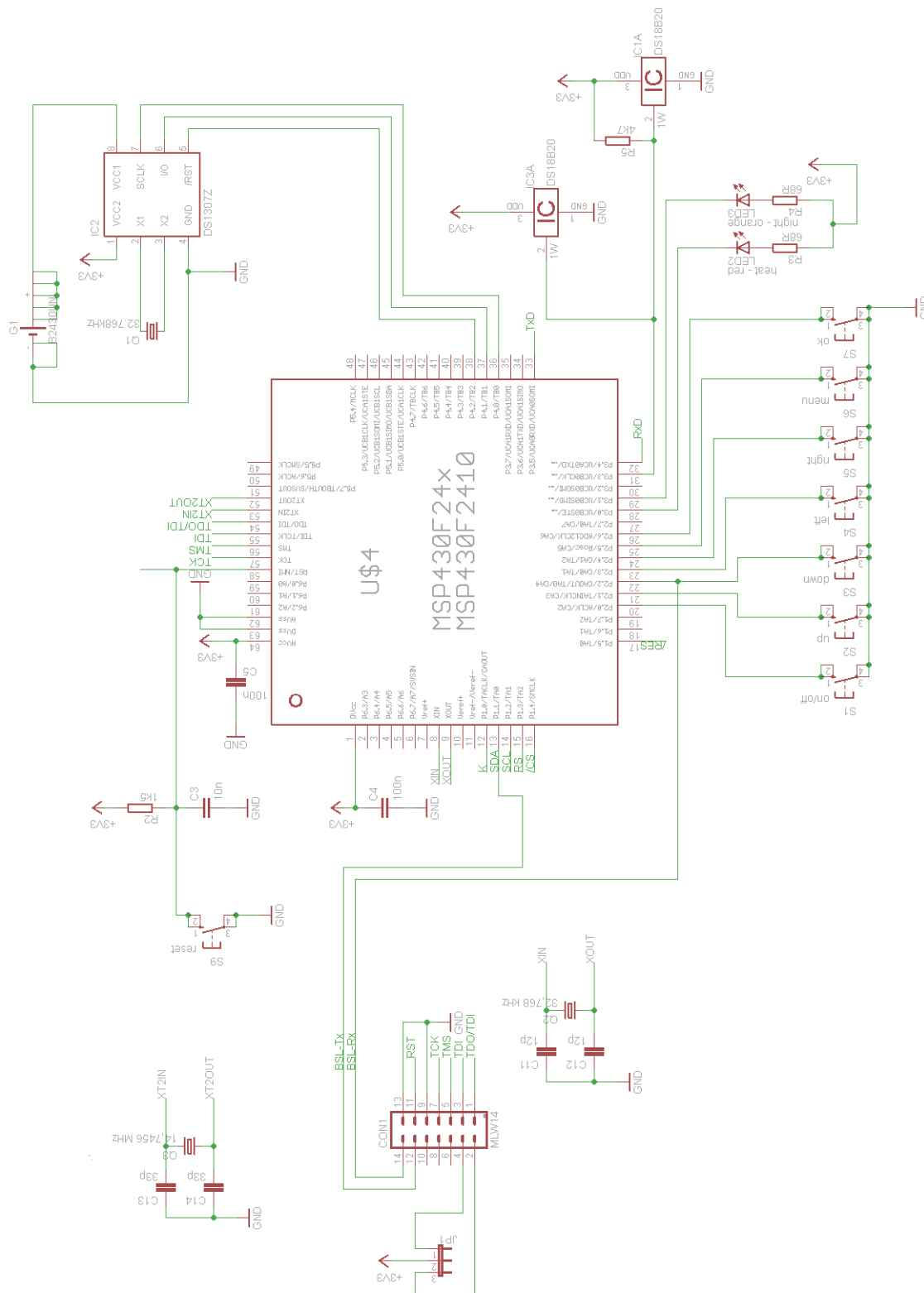
Příloha 1. Schéma zapojení termostatu navržené v programu Eagle

Příloha 2. Fotodokumentace termostatu

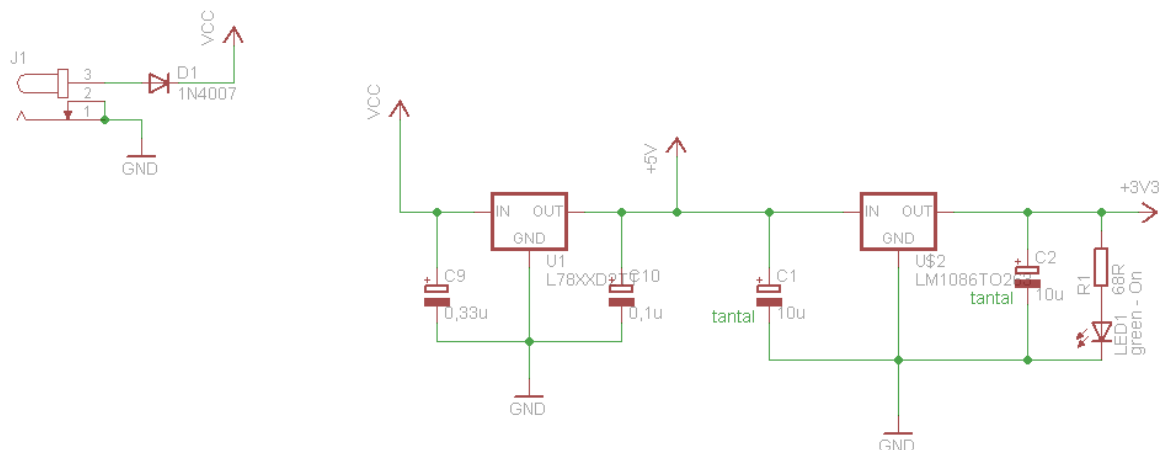
Příloha 3. Obsah CD

Příloha 1

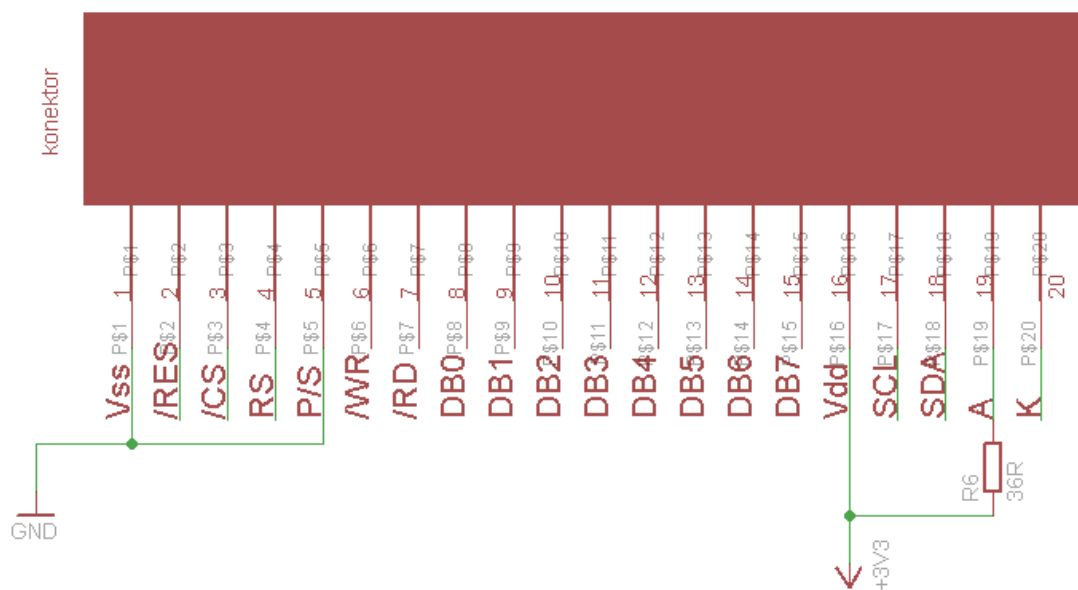
Schéma zapojení termostatu navržené v programu Eagle



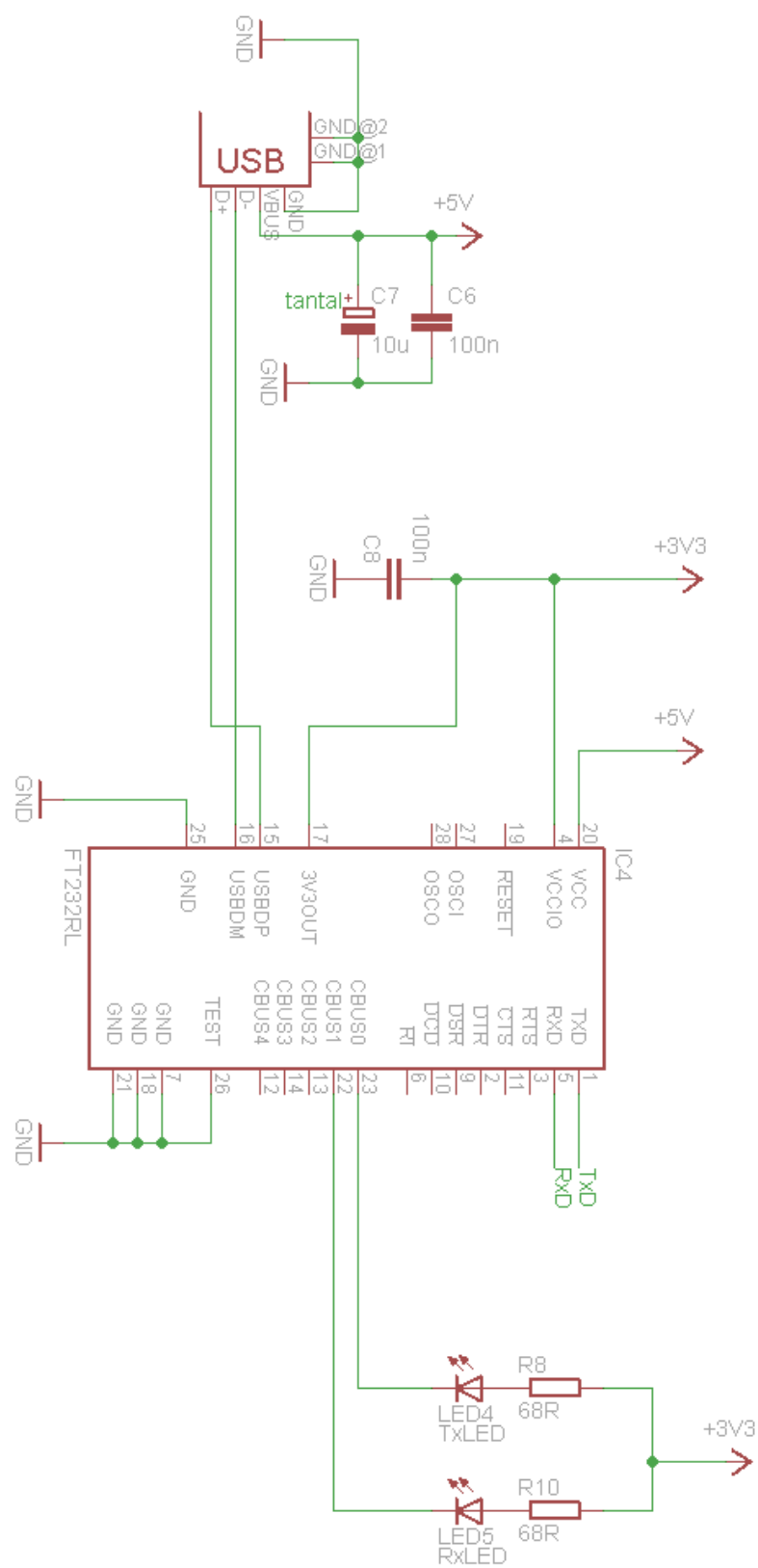
Obr. 1.1: První část schématu, zapojení tlačítek RTC a senzorů k MCU



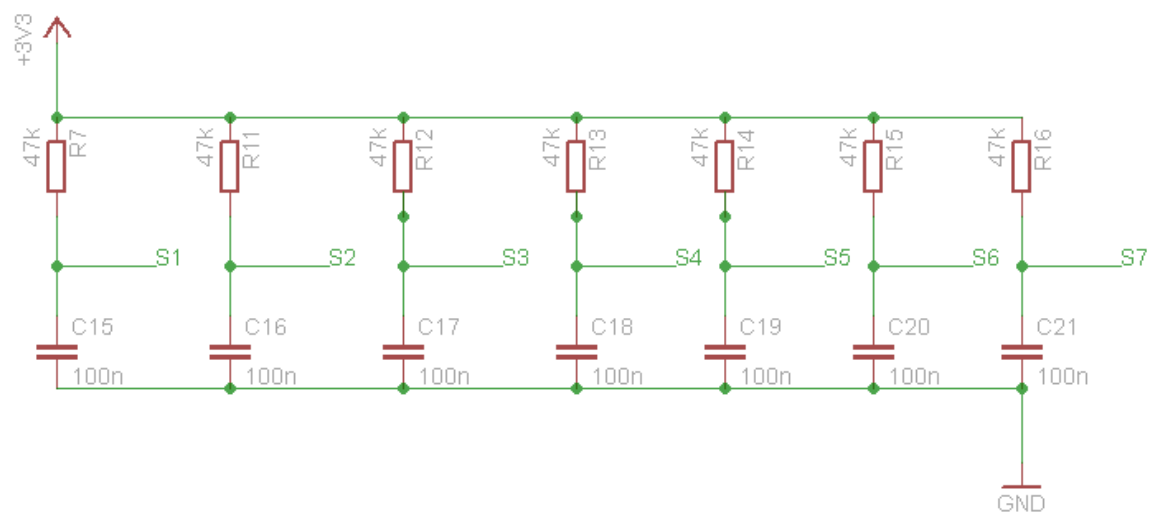
Obr. 1.2: Schéma napájecí část termostatu



Obr. 1.3: Schéma zapojení konektoru pro grafický LCD displej



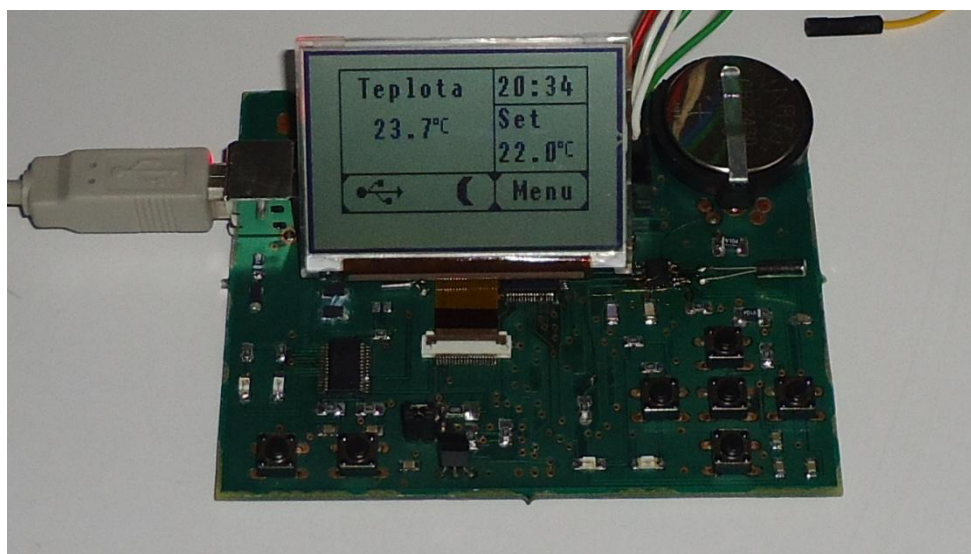
Obr. 1.4: Schéma zapojení obvodu RT232RL



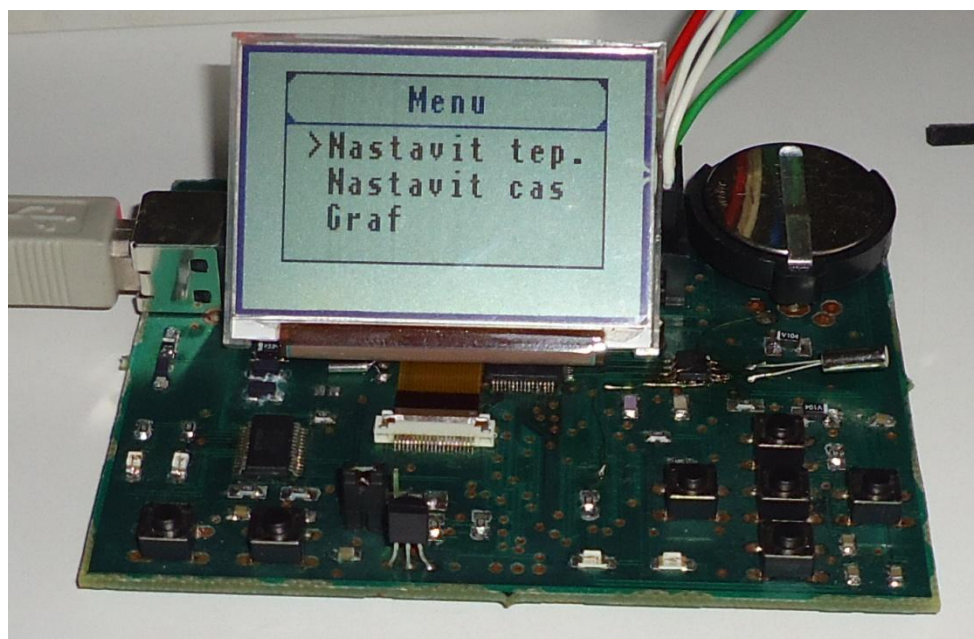
Obr. 1.5: Schéma ošetření tlačítek proti zákmitům

Příloha 2

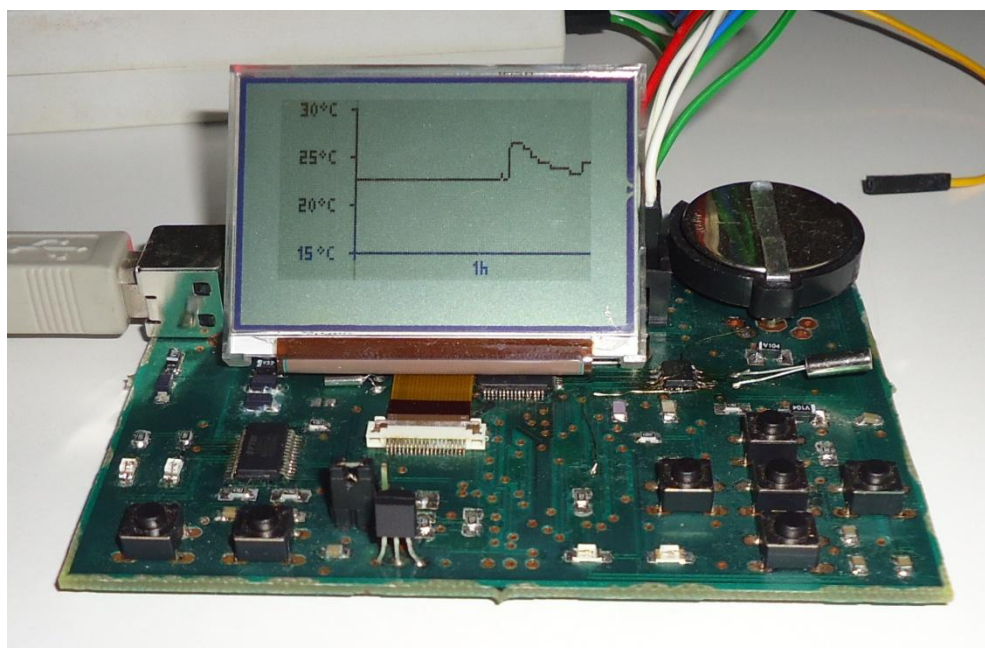
Fotodokumentace termostatu



Obr. 2.1: Hotová osazená deska termostatu s běžícím programem



Obr. 2.2: Ukázka hlavního menu



Obr. 2.3: Ukázka průběhu grafu teploty

Příloha 3

Obsah CD

- Zdrojové kódy termostatu v adresáři `/src`
- Zdrojové kódy aplikace na počítači v adresáři `/srcPc`
- Zkompilovaná aplikace pro počítač, funkční pro operační systémy Windows v adresáři `/pcApp`
- Návod pro práci s aplikací a termostatem v adresáři `/navod`
- Schéma zapojení spolu s deskou plošných spojů v adresáři `/hwNavrh`
- Tato práce ve formátu PDF v adresáři `/bpPDF`
- Tato práce ve formátu .docx spolu se všemi použitými obrázky v adresáři `/bpDOCX`