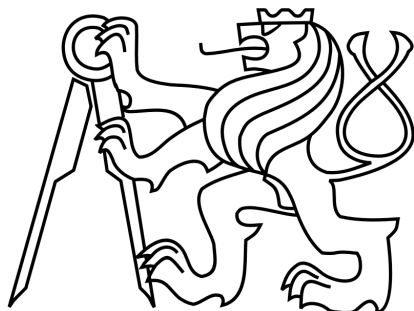


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ



DIPLOMOVÁ PRÁCE

ŘÍDICÍ SYSTÉM PRO REGULACI TEPLoty

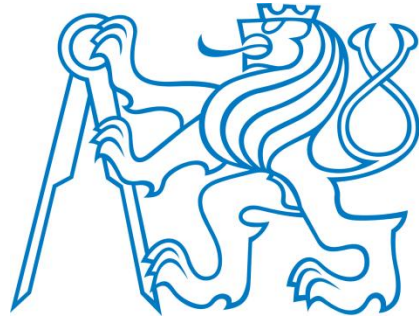
LUKÁŠ ČAREK

2008

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ

KATEDRA ŘÍDICÍ TECHNIKY



DIPLOMOVÁ PRÁCE

ŘÍDICÍ SYSTÉM PRO REGULACI TEPLoty

TEMPERATURE CONTROL SYSTEM

LUKÁŠ ČAREK

2008

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne 17.5.2008



Lukáš Čarek

Anotace

Tato práce se zabývá návrhem komponent systému pro bezdrátové řízení vytápění. Systém se skládá z 4 základních komponent, inteligentního termostatu, inteligentních akčních členů a převodníku USB - bezdrátové rozhraní. Tyto komponenty jsou navzájem propojeny pomocí bezdrátových modulů pracujících s nosnou frekvencí 868MHz. Celý systém je navržen tak, aby umožňoval maximální modularitu podle konkrétních požadavků pro daný objekt.

Klíčová slova

IRC, Nordic, MSP 430, EM6125, LPC21xx, 868MHz, FT232, řízení teploty, MSPGCC

Summary

The aim of this diploma thesis was developing of components for the wireless building temperature control system. At this moment system consists of the following components. Intelligent thermostat, intelligent actuators and USB/wireless converter. These components communicate over the wireless interface using the carrier frequency 868MHz. System was developed considering maximal modularity to satisfy requirements of the concrete building.

Keywords

IRC, Nordic, MSP 430, EM6125, LPC21xx, 868MHz, FT232, temperature control, MSPGCC

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Lukáš Čarek**

Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný
Obor: Kybernetika a měření, blok KM1 - Řídicí technika

Název tématu: **Řídicí systém pro regulaci teploty**

Pokyny pro vypracování:

Navazte na výsledky předcházející bakalářské práce a dořešte následující body:


1. Implementujte bezdrátový protokol, který umožní komunikaci mezi různými jednotkami potřebnými pro systém regulace vytápění.
2. Provedte analýzu spolehlivosti a dosahu zařízení v různých podmínkách.
3. Implementujte podporu pro externí čidla připojená na sběrnici Dallas OneWire.
4. Na základě zkušeností ze stávajícím HW proveďte potřebné úpravy zapojení a navrhnete nový plošný spoj.
5. Zvažte úpravy zapojení v případě, že místo bezdrátové komunikace bude použita sběrnice RS485.
6. Vytvořte potřebný SW pro demonstraci funkčnosti celého systému regulace vytápění (využijte případně výsledky ostatních prací, které řeší další komponenty systému).

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Pavel Němeček

Platnost zadání: do konce zimního semestru 2008/2009


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 10. 9. 2007

Obsah

Kapitola 1	Úvod	1
Kapitola 2	Současný stav IRC ve světě	2
2.1	IRC a jeho předchůdci	2
2.2	IRC a jeho podstata	3
2.3	Celkový pohled na IRC	3
2.4	IRC a komfort	4
Kapitola 3	Marketingové zamyšlení	5
3.1	Konkurenceschopnost	5
3.2	Cílové skupiny zákazníků	5
3.3	SWOT analýza	6
3.4	Finanční stránka	8
3.4.1	Bezdrátové inteligentní čidlo	8
3.4.2	Bezdrátový akční člen malý	8
3.4.3	Bezdrátový akční člen velký (ovládání kotle)	9
3.4.4	Převodník k PC + konfigurační software	9
Kapitola 4	Požadavky na jednotlivé komponenty	10
4.1	Inteligentní prostorový termostat - rfIRC25	10
4.2	Inteligentní akční člen malý - rfIRC10	12
4.3	Inteligentní akční člen velký - rfIRC15	12
4.4	Převodník USB-Nordic - rfIRC35	13
4.5	Možnosti budoucího rozšíření	13
4.6	Konfigurační software	13

Kapitola 5	Návrh zapojení jednotlivých modulů	14
5.1	Návrh zapojení bezdrátového termostatu rfIRC25	14
5.1.1	Volba součástí pro bezdrátový termostat	14
5.1.2	Zapojení bezdrátového obvodu Nordic nrf905	18
5.1.3	Zapojení pro měření napájecího napětí a měření teploty pomocí NTC.....	19
5.1.4	Zapojení měniče ST L6920	21
5.1.5	Ostatní obvody.....	22
5.2	Návrh zapojení převodníku USB-Nordic rfIRC35.....	22
5.2.1	Volba součástí pro převodník USB-Nordic	23
5.2.2	Zapojení obvodu FTDI FT232R	24
5.2.3	Ostatní obvody.....	25
5.3	Návrh zapojení malého akčního členu rfIRC10	25
5.3.1	Volba součástí pro akční člen rfIRC10.....	25
5.3.2	Zapojení napájecího zdroje.....	26
5.4	Návrh zapojení velkého akčního členu rfIRC15.....	28
5.4.1	Popis zapojení	28
5.5	Návrh zapojení programátoru pro procesory LPC21xx	29
5.5.1	Popis zapojení	29
Kapitola 6	Návrh desek plošných spojů.....	32
6.1	Návrh DPS pro bezdrátový termostat rfIRC25	33
6.1.1	Návrh obrysů desek pro ABB Time v prostředí AutoCAD	33
6.1.2	Prostorové uspořádání DPS a komponent na DPS	34
6.2	Návrh DPS pro převodník USB-Nordic rfIRC35	36
6.2.1	Návrh obrysu desky podle použité krabice.....	37

6.2.2	Prostorové uspořádání komponent na DPS.....	37
6.3	Návrh DPS pro malý akční člen rfIRC10.....	38
6.3.1	Návrh obrysů desek podle použité krabičky.....	38
6.3.2	Prostorové uspořádání DPS a komponent na DPS.....	39
6.4	Návrh DPS pro velký akční člen rfIRC15.....	41
6.4.1	Stručný popis návrhu DPS pro akční člen rfIRC15.....	41
6.5	Návrh DPS pro programátor procesorů LPC21xx.....	42
Kapitola 7	Implementace základního software.....	44
7.1	Popis knihoven pro obsluhu hardware.....	44
7.1.1	Knihovna pro obsluhu čidla DS18B20 (ds18b20.c, ds18b20.h).....	44
7.1.1	Knihovna pro interní čidlo teploty (teplomer.c, teplomer.h).....	47
7.1.2	Knihovna pro obsluhu displeje EM6125 + I2C (LCD.c, LCD.h).....	47
7.1.3	Knihovna pro obsluhu SPI rozhraní (spi.c, spi.h).....	56
7.1.4	Knihovna pro UART (usart.c, usart.h).....	57
7.1.5	Knihovna pro rozhraní reálného času (MSPClock.c, MSPClock.h).....	58
7.1.6	Knihovna pro obsluhu tlačítek (buttons.c, buttons.h).....	59
7.2	Popis nadstavbových knihoven.....	61
7.2.1	Knihovna pro obsluhu bezdrátového modulu Nordic nrf905.....	61
7.2.2	Knihovna pro funkce grafického menu (menu_lib.c, menu_lib.h).....	62
7.2.3	Knihovna pro nastavení topné křivky (tempcurve.c, tempcurve.h).....	67
Kapitola 8	Teoretický návrh bezdrátového protokolu.....	71
8.1	Struktura bezdrátové sítě.....	71
8.2	Řešení kolizí při komunikaci.....	73
8.3	Uspořádání dat v jednotlivých jednotkách.....	73

8.4 Formát přenášených dat	74
8.5 Adresace jednotek v síti	76
Kapitola 9 Reálná měření na systému.....	78
9.1 Rozbor a měření spotřeby	78
9.2 Rozbor vlastností antén a dosahu	80
9.2.1 Praktické orientační měření.....	80
Kapitola 10 Výroba krytu pro termostat rFIRC25	81
10.1 Výkresy a popis frézování krytu ABB Time	81
10.1.1 Frézování přípravku pro upevnění krytu.....	81
10.1.2 Frézování vlastního krytu.....	82
Kapitola 11 Zhodnocení.....	84
Poděkování.....	86
Použité materiály	87
Použitý software.....	88
Přílohy	89

Seznam obrázků

Obrázek 4.1 Náhled barevných designů ABB Time	11
Obrázek 5.1 Struktura mikrokontroléru MSP430 F1612	15
Obrázek 5.2 Blokové schéma obvodu Nordic nrf905	16
Obrázek 5.3 Blokové schéma čidla teploty DS18B20	17
Obrázek 5.4 Zapojení obvodu Nordic nrf905.....	19
Obrázek 5.5 Připojení děličů pro měření napětí a teploty pomocí AD převodníků.....	21
Obrázek 5.6 Zapojení měniče ST L6920.....	22
Obrázek 5.7 Zapojení obvodu FT232R	24
Obrázek 5.8 Blokové schéma mikrokontroléru LPC2103	26
Obrázek 5.9 Zapojení Step-Down konvertoru 230V ACDC/5,5V DC.....	27
Obrázek 5.10 Zapojení části pro obsluhu boot režimů procesoru LPC2103	31
Obrázek 6.1 Vzhled jednotky termostatu rfIRC25	32
Obrázek 6.2 Obrysy horní a dolní desky	34
Obrázek 6.3 Náhled DPS bezdrátového termostatu rfIRC25.....	36
Obrázek 6.4 Převodník USB-Nordic rfIRC35.....	37
Obrázek 6.5 Náhled DPS převodníku rfIRC35	38
Obrázek 6.6 Akční člen rfIRC10.....	39
Obrázek 6.7 Návrh obrysů malého akčního členu.....	39
Obrázek 6.8 Náhled DPS malého akčního členu rfIRC10.....	40
Obrázek 6.9 Krabička pro vestavbu akčního členu rfIRC15.....	41
Obrázek 6.10 Náhled DPS pro velký akční člen rfIRC15.....	42
Obrázek 6.11 USB programátor procesorů LPC21xx	42
Obrázek 6.12 Náhled DPS USB programátoru pro LPC21xx	43
Obrázek 7.1 Formát dat získaných z čidla teploty DS18B20	46
Obrázek 7.2 Náhled fontu velikosti 1 a 2.....	49
Obrázek 7.3 Ukázka vykreslení obrázku pomocí funkce <i>LcdPixel</i> ()	50
Obrázek 7.4 Náhled obrazce vytvořeného pomocí funkce <i>LcdLine</i> ().....	50

Obrázek 7.5 Použití funkce <i>LcdVLine</i> () pro rychlé kreslení vertikálních čar	51
Obrázek 7.6 Ukázka funkce <i>LcdHline</i> () pro rychlé kreslení horizontálních čar	52
Obrázek 7.7 Ukázka funkcí pro kreslení rychlých tečkovaných čar	52
Obrázek 7.8 Ukázka funkcí pro kreslení šipek	53
Obrázek 7.9 Ukázka funkce <i>LcdRectangle</i> () pro kreslení obdélníků.....	53
Obrázek 7.10 Ukázka funkce pro rychlou nonekvivalenci jedné banky.....	55
Obrázek 7.11 Schéma komponenty pro nastavení data a času	63
Obrázek 7.12 Jednotlivá okna pro editaci času a data	64
Obrázek 7.13 Nabídka pro uložení data a času	64
Obrázek 7.14 Jednotlivé položky menu	65
Obrázek 7.15 Komponenta pro zadávání číselné hodnoty.....	66
Obrázek 7.16 Náhled pro komponenty zobrazení času, dne v týdnu a data.....	66
Obrázek 7.17 Příklad graficky zadané topné křivky	68
Obrázek 7.18 Ukázka obrazovky pro zadávání teploty	69
Obrázek 7.19 Ukázka obrazovky pro zadávání času konce úseku	69
Obrázek 7.20 Ukázka potvrzovacího okna.....	70
Obrázek 7.21 Zjednodušené schéma komponenty pro nastavení topné křivky	70
Obrázek 8.1 příklad možné konfigurace systému bezdrátové regulace teploty	72
Obrázek 8.2 Příklad možnosti využití části paměti pro informace o jednotce	74
Obrázek 8.3 Struktura dat přenášených v síti.....	75
Obrázek 8.4 Princip autodetekce pomocí pulení intervalu	77
Obrázek 10.1 Výkres pro přípravu G-kódů.....	82
Obrázek 10.2 Výsledný kryt před a po úpravě.....	83

Seznam tabulek

Tabulka 3.1 SWOT analýza pro bezdrátový systém individuální regulace teploty	6
Tabulka 3.2 Konfrontační matice pro bezdrátový systém individuální regulace teploty	7
Tabulka 5.1 Převod teploty z NTC pomocí 12 bitového AD převodníku	21
Tabulka 9.1 Maximální přípustné spotřeby bezdrátového termostatu.....	78
Tabulka 9.2 Výpočet spotřeby a reálné výdrže baterií	79

Seznam příloh

Příloha A.1 Rozpiska součástek pro převodník rfIRC35	90
Příloha A.2 Rozpiska součástek pro bezdrátový termostat rfIRC25	91
Příloha A.3 Rozpiska součástek pro bezdrátový akční člen rfIRC10.....	93
Příloha A.4 Rozpiska součástek pro USB programátor procesorů LPC21xx.....	95
Příloha B.1 Schéma převodníku USB-Nordic rfIRC35	96
Příloha B.2 Schéma bezdrátového termostatu rfIRC25	97
Příloha B.3 Schéma bezdrátového akčního členu rfIRC10.....	98
Příloha B.4 Schéma USB programátoru pro procesory LPC21xx.....	99

Kapitola 1

Úvod

V současné době je jedním z nejdiskutovanějších témat úspora energií ve všech možných formách. Tato problematika dnes zasahuje i do běžných domácností, kde velkou část spotřebované energie tvoří vytápění. Zde tak vzniká prostor pro vývoj technologií, které umožní energii uspořit a navíc skloubit s potřebami uživatelů pro jejich osobní komfort a pohodlí. Tyto technologie se souhrnně označují zkratkou jako Individual Room Control (IRC), neboli individuální řízení vytápění.

Tato práce byla zpracovávána v rámci projektu CEPOT pod názvem „Komponenty bezdrátové regulace teploty“ a navazuje na předcházející bakalářskou práci s názvem „Řídící jednotka pro regulaci teploty“.

Kapitola 2

Současný stav IRC ve světě

Jak již z názvu IRC vyplývá, jde o individuální vytápění jednotlivých místností, tato koncepce umožňuje nastavit individuální topné režimy pro každou místnost, ale zároveň je možné celý systém ovládat z jednoho místa. Toto je hlavní rozdíl od běžně prodávaných prostorových termostatů. IRC v sobě ale přináší nové možnosti, z nichž nejdůležitější je možnost zásahu do řízení centrálního kotle.

IRC však není jen o úsporách na vytápění, ale také o celkovém komfortu pro uživatele objektu.

2.1 IRC a jeho předchůdci

Prvními předchůdci IRC systémů byly klasické termostaty, které na základě teploty a předepnutí bimetalového pásku spínaly či odpínaly přímotopná tělesa. V případě teplovodního topení se jednalo o tzv. termohlavice. Hlavice jsou založeny na principu teplotní roztažnosti média kterým je vyplněn vlnovec. Tento vlnovec pak na základě změny teploty mění svoji délku a tím přivírá či otevírá ventil radiátoru. Princip obou předchozích zařízení v sobě integruje jak snímání teploty tak akční člen.

V současné době se stále více začínají používat elektronické prostorové termostaty, které již umožňují zavést časové změny pro řízení jednotlivých pokojů. Počet změn bývá často omezen pouze na 2 změny denně a navíc tyto termostaty mají často možnost nastavit topný režim pouze pro pracovní dny a víkend.

Dnes se začínají v sítích supermarketů a hobbymarketů objevovat i bezdrátové termostaty spárované s daným akčním členem. Toto řešení umožňuje měřit teplotu mimo místo které je ovlivněno zdrojem tepla, a tudíž nastavit i lepší tepelnou pohodu pro danou místnost. Zároveň také, na rozdíl od klasických termostatů, umožňuje instalaci bez větších zásahů do místnosti. Opět zde ale bývají omezení v podobě počtu denních změn a neúplném týdenním programu.

2.2 IRC a jeho podstata

Jak již bylo napsáno výše hlavní přínos je v možnosti centrálního spravování topných programů pro jednotlivé místnosti a možnost koncentrovat informace o teplotě případně dalších fyzikálních veličin, jako například venkovní teplota či rychlost a směr větru, ovlivňující nějakým způsobem vytápění do jednoho místa. S touto informací lze dále pracovat a lze pomocí ní ovlivnit centrální zdroj tepla (kotel) a zvýšit tak efektivitu celého systému vytápění.

2.3 Celkový pohled na IRC

Systémy individuálního vytápění jsou ideální volbou jak pro nově postavené domy, zde se uplatní zejména klasické IRC systémy komunikující po drátové sběrnici kde odpadá problém s napájením jednotlivých komponent celého systému. Pro rekonstrukce se hodí spíše bezdrátové systémy které umožní změnu stávajícího systému vytápění za IRC bez nutnosti hlubších zásahů do budovy.

2.4 IRC a komfort

Každý dům má minimálně dvě, spíše tři a více částí, s různými nároky na vytápění. Tyto nároky jsou děleny podle typu a použití daného pokoje a zároveň podle denní doby, ve kterou je používán. Například pro obývací pokoj je ideální teplota kolem 21°C v časovém rozmezí od 15 do 16 hodin, kdy se po pracovní době vracíme domů a užíváme zasloužený odpočinek, dále je užíván až kolem 20 až 23 hodiny. Právě v těchto časových intervalech je požadována správná teplota a mimo tyto časové úseky může být nižší. Dalším příkladem místnosti, kde není třeba konstantní teplota po celý den je koupelna. Zde je optimum 23°C v ranních a večerních hodinách. Při kancelářských pracích je vhodné nepřetápět, a tak teplota může být málo nad 19°C, ale opět v kanceláři, či místnosti, kterou používáte jako pracovnu stačí takováto teplota pouze v pracovních hodinách a mimo ně nižší. IRC umožňuje i týdenní plány a tak například pro víkend mohou být tyto teploty rozdílné než pro běžné pracovní dny.

Kapitola 3

Marketingové zamyšlení

Na trhu se již dnes objevuje celá spousta výrobců IRC systému, a případný uživatel má tak téměř neomezený výběr. Ovšem i tento výběr je do značné míry limitován finančními prostředky a tak zejména mezi levnějšími systémy je problém vybrat systém který je uživatelsky přátelský.

3.1 Konkurenceschopnost

Aby tento systém při příchodu na trh byl konkurenceschopný, musí nabízet „něco navíc“ za srovnatelnou nebo nižší cenu. Dalším cílem je oslovení co největší skupiny potenciálních zákazníků (distributorů).

3.2 Cílové skupiny zákazníků

Hlavní využití je tedy plánované hlavně mezi rodinnými domy, penziony či malými administrativními celky. Jelikož při rekonstrukcích či nových projektech obytných jednotek, je vytápění samostatná kapitola u které je nutné zpracovat projekt

podle konkrétních prostor a požadavků, je jako cílová skupina brán v úvahu distributor, zabývající se projektováním takovýchto systémů.

3.3 SWOT analýza

Tabulka 3.1 ukazuje síly(**Strengths**), slabosti(**Weaknesses**), příležitosti(**Opportunities**) a hrozby(**Threats**) pro IRC systém bezdrátové regulace teploty

	Positive	Negative
Internal	<p>Strengths</p> <p>S1. kvalita technického řešení a vysoká úroveň zpracování S2. mnoho barev a designů S3. dlouhá výdrž baterií S4. grafické rozhraní S5. možnost externích LowCost čidel S6. nouzová regulace při poruchách S7. nezávislost na kabelech</p>	<p>Weaknesses</p> <p>W1.omezené množství produkce W2.nedostatečné finanční zdroje W3.problémy funkcí v železobetonových stavbách W4.slabé jméno v oblasti domácí automatizace</p>
External	<p>Opportunities</p> <p>O1.distribuce na zahraniční trhy O2.vyšší zájem o rekonstrukce O3.tlak na ekonomiku vytápění O4.nové zdroje pro části zařízení vedoucí ke zlevnění eventuelně k vylepšení O5.lepší se jméno českých produktů v EU</p>	<p>Threats</p> <p>T1. příchod levných alternativ z Číny T2. lobismus ze strany distributorů T3. zákonné omezení ohledně používání bezlicenčního pásma 868MHz T4. úmyslně poškozené jméno produktu T5. úmyslné snížení cen konkurence za účelem nempustit na trh další firmu</p>

Tabulka 3.1 SWOT analýza pro bezdrátový systém individuální regulace teploty

Z těchto podkladů je možné sestavit konfrontační matici (Tabulka 3.2) ze které jsou vidět největší klady a největší zápory při uplatnění systému na trhu.

		Strengths			Weaknesses			
		S1. Kvalita	S7. Bezdrátové	S4. Grafické rozhraní	W2. Finanční zdroje	W2. Železobetonové stavby	W4. Slabé jméno	
Opportunities	O1. Zahraničí	++	++	++	-	0	--	+ 3
	O2. Rekonstrukce	+	+++	+	0	--	-	+ 2
	O3. Vláda	0	0	0	+	0	0	+ 1
Threats	T1. Čína	-	+	+	0	0	0	+ 1
	T3. Zákon	0	---	0	-	---	0	- 7
	T5. Konk. boj	+++	++	+++	--	--	---	+ 1
		+ 5	+ 6	+ 7	- 4	- 7	- 7	

Tabulka 3.2 Konfrontační matice pro bezdrátový systém individuální regulace teploty

Tabulka 3.2 ukazuje, že nejvyšší působnosti na trhu je možné dosáhnout moderními technologiemi jako je bezdrátové řízení a kvalitní ergonomické prostředí. Naopak je zde několik negativních faktorů, které by mohly ohrozit vstup na trh. Pro bezdrátový systém by byl velký problém v omezení bezlicenčního pásma 868MHz, špatná funkce nebo úplná nefunkčnost v železobetonových stavbách, případně slabé jméno v oblasti technického zařízení budov.

Ze SWOT analýzy je patrné, že v případě nasazení systému do prodeje by bylo dobré, aby existovala varianta pracující na klasické drátové sběrnici, kde bylo možné učinit nabídku i zákazníkům železobetonových staveb novostaveb za mírně nižší cenu než je bezdrátový systém. Zároveň by tento přístup pomohl vyzdvihnout případné jméno produktu v podvědomí veřejnosti.

3.4 Finanční stránka

V této části je uveden přehled předpokládaných nákladů na vývoj, výrobu a distribuci každého modulu, který zahrnuje ceny od vývoje až po koncový prodej zákazníkovi. Nástřel ceny je proveden na základě průzkumu konkurenčního trhu, s přihlédnutím k možnostem vyvíjeného systému.

3.4.1 Bezdrátové inteligentní čidlo

Výroba elektroniky (hardware)	700 Kč
Obal, návod, balení	70 Kč
Náklady na vývoj	100 Kč
Náklady na propagaci	100 Kč
Marže 30% pro distributora	525 Kč
Koncová prodejní cena	1750 Kč
Zisk	255 Kč

3.4.2 Bezdrátový akční člen malý

Výroba elektroniky (hardware)	350 Kč
Obal, návod, balení	50 Kč
Náklady na vývoj	100 Kč
Náklady na propagaci	100 Kč
Marže 30% pro distributora	345 Kč
Koncová prodejní cena	1150 Kč
Zisk	205 Kč

3.4.3 Bezdrátový akční člen velký (ovládání kotle)

Výroba elektroniky (hardware)	700 Kč
Obal, návod, balení	50 Kč
Náklady na vývoj.....	100 Kč
Náklady na propagaci	100 Kč
Marže 30% pro distributora.....	870 Kč
Koncová prodejní cena	2900 Kč
Zisk.....	1030Kč

3.4.4 Převodník k PC + konfigurační software

Výroba elektroniky (hardware)	200 Kč
Obal, návod, balení	80 Kč
Náklady na vývoj.....	400 Kč
Náklady na propagaci	100 Kč
Marže 30% pro distributora.....	1350 Kč
Koncová prodejní cena	4500 Kč
Zisk.....	2370 Kč

Uvedené ceny platí samozřejmě až pro sérii, čítající kolem 500 Ks od každého modulu.

Kapitola 4

Požadavky na jednotlivé komponenty

Celý bezdrátový systém by měl sestávat ze čtyř základních komponent, které by bylo v budoucnu možno doplnit i o další jednotky.

1. Inteligentní prostorový termostat.....rfIRC25
2. Inteligentní akční člen malý.....rfIRC10
3. Inteligentní akční člen velký..... rfIRC15
4. Převodník USB – Nordic..... rfIRC35

4.1 Inteligentní prostorový termostat – rfIRC25

Jak již název komponenty napovídá jedná se o termostat který bude sloužit k měření teploty a následné regulaci v páru „termostat-akční člen“. Zároveň tato jednotka musí být v jistých časových okamžicích dostupná i pro ostatní komponenty systému (například pro konfiguraci či změnu topné křivky z jiného místa).

Aby bylo možné využít bezdrátový systém se všemi jeho výhodami, je nutné, aby tyto jednotky bylo možné napájet z baterií. Dále je žádoucí aby jednotky měly možnost

napájení i z externího zdroje pro případ použití parazitního napájení například ze spínače osvětlení.

Pro možnost vytvoření nízkorozpočtového systému (tzv. LowCost) je žádoucí, aby každý termostat měl možnost připojení i nějakého externího čidla (např. pro využití jednoho termostatu pro dvě sousedící místnosti. Pro speciální užití je žádoucí vybavení termostatu ještě čidlem vlhkosti (typické použití například v koupelně).

Pro vytvoření příjemného ergonomického ovládání je potřeba osazení jednotky plnohodnotným grafickým displejem, pomocí něhož bude možné například nastavit topný program přímo na termostatu.

V tomto případě se jedná o jednotku, která bude součástí interiéru daného objektu a je tedy nutné vzít v potaz i design takového termostatu a je tedy nutné řešit zástavbu do krabičky. S přihlédnutím k možnosti včlenění mezi ostatní elektroinstalaci, jako například spínače osvětlení či zásuvkové kryty, byla zvolena krabička Time od firmy ABB. Jedná se o krabičku rozměrů nástěnného spínače osvětlení, která je vyráběna v celé škále barev viz. Obrázek 4.1



Obrázek 4.1 Náhled barevných designů ABB Time

4.2 Inteligentní akční člen malý – rfIRC10

Jedná se o jednotku s jedním spínacím a jedním rozpínacím kontaktem, která může sloužit například k ovládání přímotopů, podlahového topení, eventuelně pro ovládání radiátorového ventilu v případě regulace objektu s teplovodními vytápěním.

Pro požadavek nezávislosti bezdrátového systému je nutné, aby tento akční člen měl svůj nezávislý napájecí zdroj, který umožní napájení přímo z elektrické sítě. Zároveň by měl umět spínat relativně vysoký výkon aby bylo vyhověno požadavku na spínání přímotopných těles.

Jelikož se jedná o zařízení kterých v systému bude relativně velké množství, měl by být brán ohled na co nejnižší cenu.

Poslední z požadavků na tento akční člen jsou malé rozměry, aby byla možná vestavba i do nízké přístrojové krabice do sádkokartonu, případně umístění pod spínač osvětlení.

4.3 Inteligentní akční člen velký – rfIRC15

Tento akční člen by měl sloužit k ovládání globálního zdroje tepla pro daný objekt (kotel, tepelné čerpadlo atd.), proto by měl být vybaven alespoň 4 silovými výstupy, digitálními i analogovými vstupy a digitálním výstupem pro možnost implementace nějakého komunikačního prostředku (např. protokol OpenTherm).

Jelikož se jedná o jednotku, která nebude primárně určená pro zástavbu do interiéru, není zde nutné klást důraz na estetiku či rozměry.

Kvůli možnosti připojení vstupů s bezpečným malým napětím, je žádoucí aby existovalo galvanické oddělení této nízkonapěťové části od síťového napájecího napětí.

4.4 Převodník USB-Nordic – rfIRC35

Jedná se o jednotku, která bude zprostředkovávat, komunikaci mezi nadřazeným systémem (konfigurační aplikace v prostředí .NET, spuštěná na běžném domácím PC), a systémem bezdrátové regulace teploty.

Tato jednotka by mohla být zprostředkovatelem událostí, které by mohly být dále navázané na chod systému (např. událost typu „poslední uživatel opustil dům“). Proto by měla mít možnost napájení z externího zdroje nezávislého na napájení z PC

Převodník bude připojen k PC pomocí sběrnice USB, ale měla by být zachována i podpora linky RS232. U této jednotky není nijak zvlášť velký požadavek na velikost proto je třeba zvolit kryt s ohledem na estetiku.

4.5 Možnosti budoucího rozšíření

V budoucnu by bylo možné systém rozšířit i o další moduly jakými jsou např. venkovní čidla, či miniaturní moduly okenních kontaktů, eventuelně bezdrátové spínače (vznik události např. poslední uživatel domu stiskne tlačítko před opuštěním – vznik události pro systém)

4.6 Konfigurační software

Pro správnou činnost je třeba systém IRC nakonfigurovat, proto je potřeba speciální software, který umožní nastavení příslušných vazeb mezi jednotlivými komponentami a dále bude uživateli umožňovat snadnou změnu parametrů topných programů, vytváření nových atd. Tento software bude vyvíjen odděleně mimo tuto práci.

Kapitola 5

Návrh zapojení jednotlivých modulů

V této kapitole bude vysvětlena volba součástek pro jednotlivé moduly a budou dopodrobna probrány jednotlivé části schémat pro konkrétní jednotky. Pro návrh bylo použito vývojové prostředí OrCAD Capture CIS v10.3. Všechna schémata jsou součástí příloh.

5.1 Návrh zapojení bezdrátového termostatu rfIRC25

Jelikož se jedná o jednotku která umožňuje napájení z baterií, je nutné k tomuto faktu přihlížet. Do zvolené krabičky ABB Time se navíc vejdou pouze dva AAA články, z čehož plynou extrémní nároky na nízkou spotřebu celé jednotky.

5.1.1 Volba součástek pro bezdrátový termostat

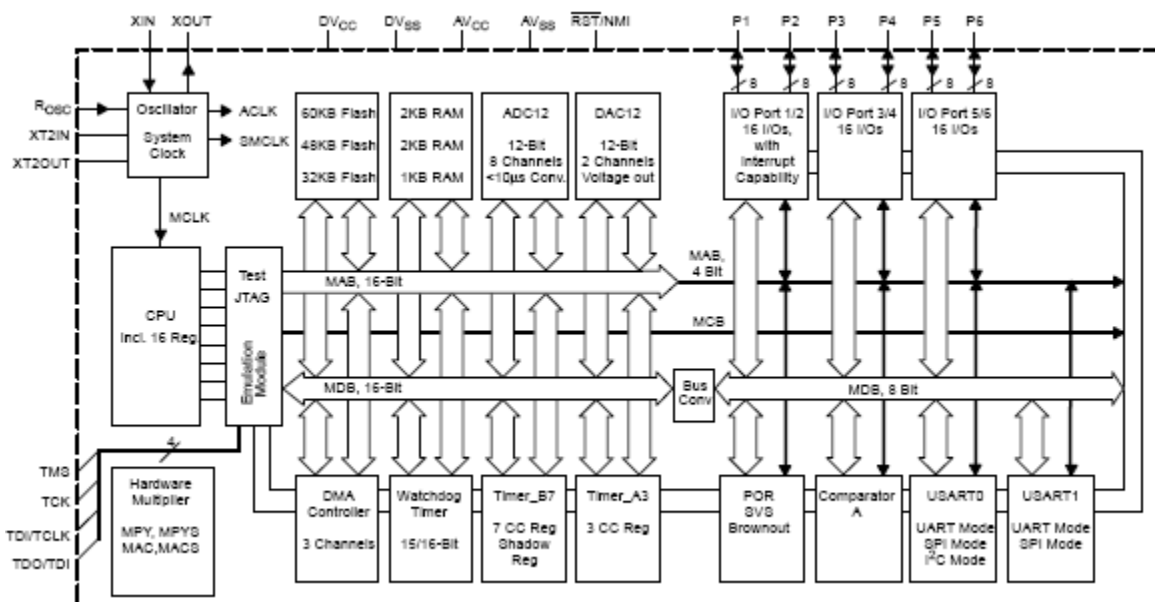
Jelikož tato práce navazuje částečně na bakalářskou práci, ve které byla řešena řídicí jednotka pro regulaci teploty, byly k volbě komponent pro danou jednotku již jisté zkušenosti. Soupis součástek použitých na bezdrátovém termostatu, je součástí přílohy.

Mikroprocesor MSP430

Protože se musí jednat o procesor, který má relativně rozumné výpočetní schopnosti, ale zároveň je kladen velký důraz na jeho spotřebu, byl zvolen procesor MSP430 F167.

Jedná se o 16 bitový mikroprocesor von Neumannovy architektury, vybavený 2kB RAM paměti a 32kB flash paměti, který má hardwarovou podporu důležitých sběrnic jako je I2C či SPI. Při vývoji software se zjistilo, že procesor nedostačuje ani velikostí RAM ani velikostí flash paměti a byl vyměněn za pinově kompatibilní F1612 s 5kB RAM a 55kB flash paměti. Procesor je navíc vybaven tříkanálovým DMA. Vnitřní strukturu procesoru ukazuje Obrázek 5.1

Mezi další přednosti mikrokontrolérů MSP430 patří například velmi propracované hodinové moduly, umožňující spolupráci s několika úspornými módy

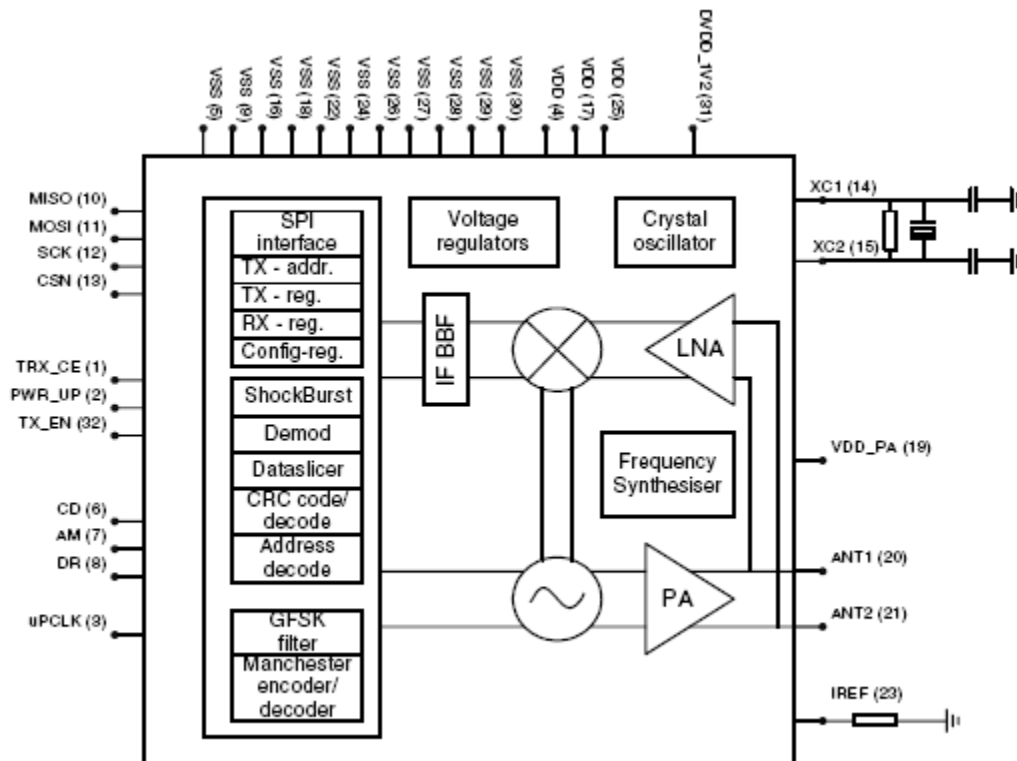


Obrázek 5.1 Struktura mikrokontroléru MSP430 F1612

Bezdrátový obvod Nordic nrf905

Při volbě bezdrátového obvodu bylo zvažováno několik možností které jsou v současné době k dispozici. Ohled byl brán opět hlavně na spotřebu a proto padla volba na obvody Nordic, které jsou vyhlášené svou technologií SchockBurst™, která je navržena tak, aby bylo možné snížit spotřebu v kritickém okamžiku, kterým je odesílání dat. V tomto

okamžiku je třeba relativně velký odebíraný výkon ze zdroje energie, proto tyto obvody využívají přenosovou rychlost vhodně zvolenou tak, aby integrál výkonu v čase byl co nejnižší. Tento obvod je navržen pro 3 pásma 433MHz, 868MHz a 915MHz. Poslední jmenované pásmo však v Evropě není povoleno. Nordic nrf905 také řeší část linkové vrstvy jako je například CRC či hlavička přenášených dat včetně adresování. Toto by mohla být do jisté míry nevýhoda pro možnost budoucí modularity či spolupráce systému s konkurenčními výrobky, avšak užití stejné fyzické adresy pro všechny komponenty daného systému umožní zpracování dat až na úrovni mikrokontroléru. Tato problematika je hlouběji probrána v Kapitola 8 týkající se bezdrátového protokolu. Blokové schéma obvodu Nordic nrf905 ukazuje Obrázek 5.2



Obrázek 5.2 Blokové schéma obvodu Nordic nrf905

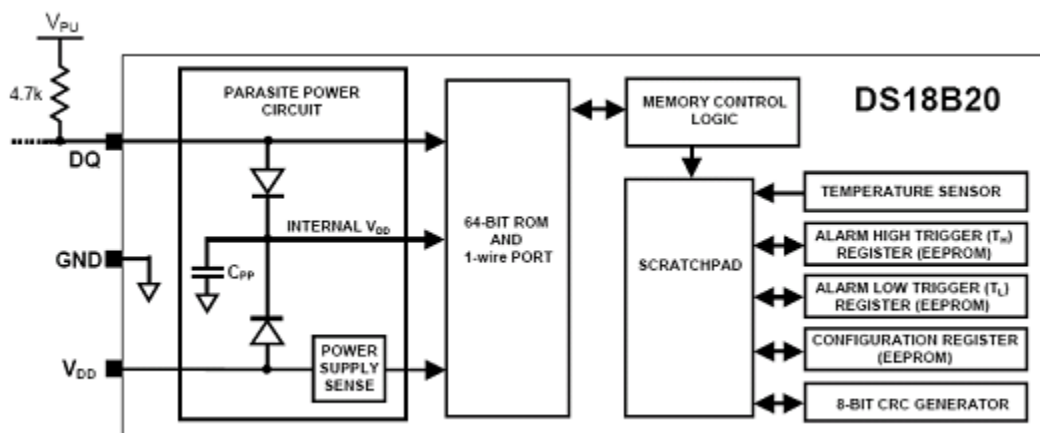
Grafický displej EM6125

Tento displej je jeden z mála grafických displejů, splňujících požadavky na nízkou spotřebu, rozumné rozlišení a zároveň díky svým rozměrům umožňuje zakomponování do použité krabičky ABB Time. Jedná se o displej typu „ChipOnGlas“ který má

integrováný řadič, včetně zdroje střídavého napětí. Další nespornou výhodou je velmi malý počet pasivních součástek potřebných pro funkci. Displej komunikuje pomocí rozhraní I2C a umožňuje také několik úsporných módů, v nichž je možné snížit spotřebu až na polovinu oproti základnímu módu.

Čidlo teploty Dallas DS18B20

Jde o číslicové čidlo teploty s 12-bitovým převodníkem. Čidlo komunikuje pomocí sběrnice 1-Wire. Tuto sběrnici podporují pouze procesory od firmy Maxim a podporu této sběrnice v procesoru MSP430 je nutno řešit softwarově. Čidlo pracuje při teplotním rozsahu od -55°C do $+125^{\circ}\text{C}$. Přesnost čidla je $0,5^{\circ}\text{C}$ v rozsahu od -10°C do $+85^{\circ}\text{C}$. Nejnižší rozlišitelný teplotní rozdíl je $0,0625^{\circ}\text{C}$. Blokovou strukturu ukazuje Obrázek 5.3



Obrázek 5.3 Blokové schéma čidla teploty DS18B20

Čidlo teploty a vlhkosti Sensirion SHT10

Jde o kalibrované čidlo vlhkosti a teploty s digitálním výstupem. Každé čidlo má své vlastní kalibrační konstanty uloženy ve své paměti. Teplota a vlhkost je ze senzorů převáděna pomocí 14-bitového převodníku a umožňuje tak velmi jemné vyčítání obou veličin. Senzor je připojen pomocí jednoduchého sériového rozhraní, které rovněž není podporováno procesorem MSP430.

5.1.2 Zapojení bezdrátového obvodu Nordic nrf905

Vzhledem, k faktu, že se jedná součástku, pracující ve vysokofrekvenčních pásmech, velmi záleží na volbě součástek. Pro aplikaci tohoto obvodu v systému inteligentní regulace teploty byla zvolena pracovní frekvence 868MHz, která má výhody oproti běžně používané frekvenci 433MHz dvě nesporné výhody:

1. Velikost antény - jelikož se jedná o zástavbu do relativně malé krabičky, je žádoucí aby použitá anténa nevycházela příliš velká.
2. Rušení - v pásmu 433MHz v současné době pracuje spousta zařízení jako například bezdrátové domovní zvonky, či běžně prodávané domovní meteostanice s bezdrátovými senzory teploty.

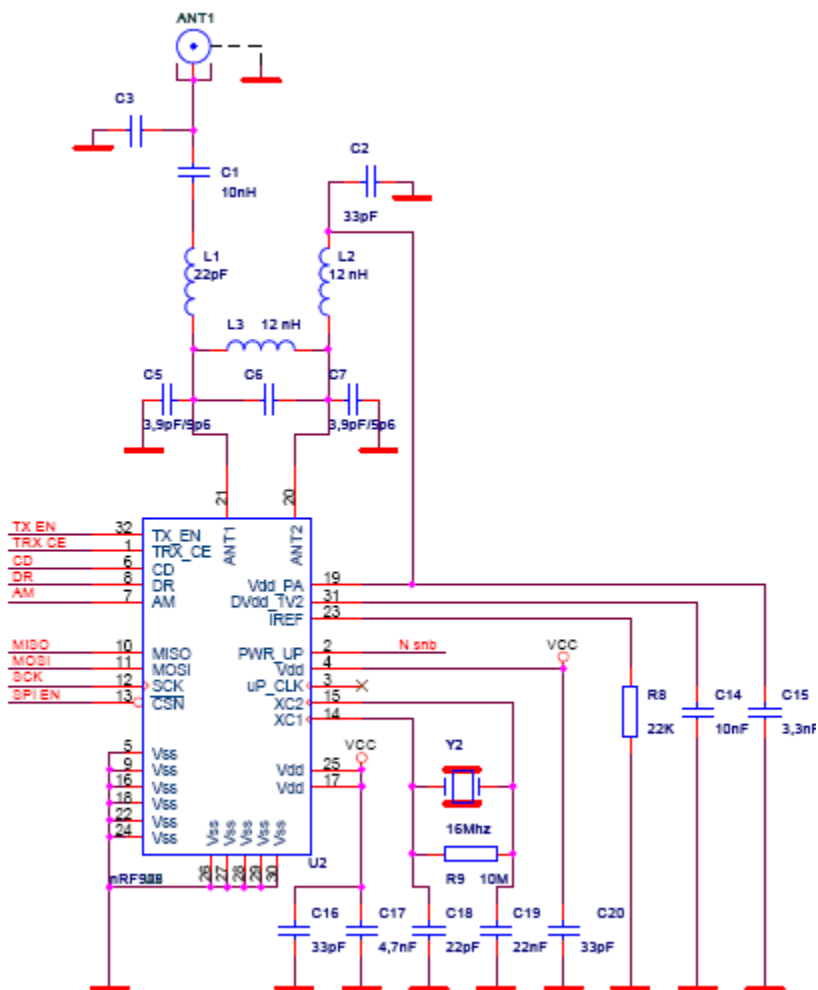
Použité hodnoty pasivních součástek jsou použity z katalogu. Schéma připojení obvodu ukazuje Obrázek 5.4.

Obvod je k procesoru připojen pomocí datové sběrnice SPI, pomocí níž probíhá nahrávání inicializačních a konfiguračních dat do registrů obvodu. Samotné ovládání příjmu vysílání a ostatních módů obvodu je realizováno pomocí několika řídicích signálů, díky nimž je přednastavení velice rychlé.

- TX_EN - (vstupní) pokyn pro odeslání dat nahraných v TX registru obvodu
- TRX_CE - (vstupní) přepne obvod do režimu příjmu
- CD - (výstupní) signál je nastaven na logickou 1 pokud obvod nrf905 detekuje nosnou vlnu na správném kanálu toto je nutný signál pro budování bezdrátových sítí, aby se zmenšila pravděpodobnost možných kolizí.
- AM - (výstupní) signál je nastaven na logickou 1 pokud obvod přijímá data s fyzickou adresou jemu vlastní. Tzn. pomocí tohoto signálu je možné upozornit mikrokontrolér, že brzy budou platná data
- DR - (výstupní) obvod nrf905 přijal datový paket s platnou fyzickou adresou a správným CRC kódem. V okamžiku kdy je tento signál v logické 1, je možné vyčíst platná data z RX registru. Další funkce tohoto signálu je také potvrzení

odeslaných dat, kdy přejde do úrovně log 1 v okamžiku, kdy se obvodu podařilo odeslat celý datový paket.

- N_snb - (vstupní) tento signál uvede obvod do režimu spánku, a umožní tak snížit spotřebu.



Obrázek 5.4 Zapojení obvodu Nordic nrf905

5.1.3 Zapojení pro měření napájecího napětí a měření teploty pomocí NTC

Mikrokontrolér MSP403 F1612 je vybaven supervizorem pro hlídání napájecího napětí. Toto je dobré pro určení zda je už napájecí napětí pod danou úrovní, ale například pro zjištění velikosti napájecího napětí, které je možné pomocí tabulky převést na zbývající kapacitu je tato funkce velice nepraktická. Měření pomocí AD převodníku je

problematické, protože je problém s napěťovou referencí. Když uvážíme minimální napájecí napětí 2V, se kterým je jednotka ještě schopna pracovat, lze použít pouze interní napěťovou referenci 1,5V. Z toho ale vyplývá, že maximální napětí, které je pomocí AD převodníku převést na číslo je právě napětí reference. Z tohoto důvodu je nutné pro měření napětí použít napěťový dělič. Toto je ovšem v protikladu s malým odběrem, pokud by totiž děličem protékal i proud kolem desítek μA vedlo by to ke zhoršení výdrže baterií. Proto je tento dělič připojen nikoliv přímo na napájecí napětí, ale na bránu procesoru. Při rozumně zvolených hodnotách rezistorů děliče bude napětí na bráně rovno napájecímu a pomocí AD převodníku připojeného na střed děliče lze získat hodnotu napájecího napětí.

Toto řešení je použito i u měření teploty pomocí NTC rezistoru. Pro měření odporu je třeba zdroj konstantního proudu, tento zdroj je však problém a další součástka navíc, která by zvyšovala cenu celého zařízení. Pro zjištění teploty je tedy použit odpor $33k\Omega$ který spolu s NTC $10k\Omega$ tvoří dělič. Pro naši aplikaci budeme uvažovat teplotní rozsah od $-10^{\circ}C$ do $+60^{\circ}C$. Pro tento teplotní rozsah udal výrobce hodnotu koeficientu R_t/R_{25} kde R_{25} je hodnota NTC termistoru při $25^{\circ}C$ a R_t je hodnota při dané teplotě t . Hodnota NTC termistoru byla zvolena tak, aby bylo pomocí 12 bitového AD převodníku možno získat relativně přesnou informaci o teplotě. Tabulka 5.1 ukazuje jak bude pro dané teploty vypadat převod na napětí a následně pomocí AD převodníku na číslo. Z této tabulky je také vidět nelinearita a abychom mohli dopočítat i mezilehlé hodnoty, je třeba užít nějaké metody interpolace. Vzhledem k relativně jemnému odstupňování jednotlivých teplot (po $5^{\circ}C$), je možné užít lineární interpolaci a chyba v daném úseku bude zanedbatelná.

Obrázek 5.5 ukazuje připojení děličů k mikrokontroléru. Signál Vdd1 je výstupní úroveň z pinu P6.2, signál Vdd je vstup do AD převodníku na pinu P6.3. Podobná je i situace s měřením napětí na NTC, signál NTC1 je výstupní úroveň z pinu P6.4 a signál označený NTC je vstup do AD na pinu P6.6

Princip měření teploty je následující:

- Pomocí prvního děliče dojde ke změření přesné hodnoty napájecího napětí.
- Tato hodnota je stejná jako napájecí napětí děliče s NTC a lze ji zpětně užít pro výpočet správné hodnoty napětí při převodu teploty z NTC termistoru



Obrázek 5.5 Připojení děličů pro měření napětí a teploty pomocí AD převodníků

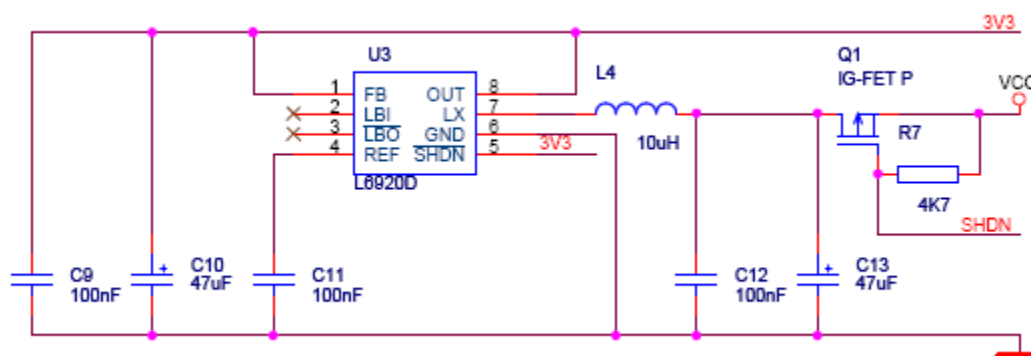
T (°C)	R_t/R_{25} (-)	R_t (k Ω)	Napětí na děliči (V)	Hodnota AD po převodu (-)
-10	3,5432	35,432	1,2944	3534
-5	2,9035	29,035	1,1701	3195
0	2,3950	23,950	1,0513	2870
5	1,9880	19,880	0,9398	2566
10	1,6602	16,602	0,8367	2284
15	1,3944	13,944	0,7425	2027
20	1,1777	11,777	0,6575	1795
25	1,0000	10,000	0,5813	1587
30	0,8534	8,534	0,5136	1402
35	0,7319	7,319	0,4538	1239
40	0,6307	6,307	0,4011	1095
45	0,5459	5,459	0,3548	969
50	0,4746	4,746	0,3143	858
55	0,4143	4,143	0,2788	761
60	0,3631	3,631	0,2478	676

Tabulka 5.1 Převod teploty z NTC pomocí 12 bitového AD převodníku

5.1.4 Zapojení měniče ST L6920

Protože je termostat primárně určen pro napájení z baterií, které mají nestále výstupní napětí, je třeba pro správnou funkci některých součástí jako čidla teploty DS18B20 či

čidlo vlhkosti SHT10 napájecí napětí 3,3V. Proto je termostat vybaven STEP-UP konvertorem L6920, který je schopen s vysokou účinností dávat výstupní napětí 3,3V z napájecího napětí od 1,1V. Problém tohoto obvodu v aplikacích napájených z baterií je, že neobsahuje tzv. funkci TrueShutDown, která při usnutí obvodu zároveň odpojí výstup tak, že jím neprotéká žádný proud. Tato nectnost je dána parazitní diodou spínacího MOS tranzistoru na výstupu měniče. Aby bylo možné omezit odběr proudu, byla funkce usnutí vynechána a obvod je odpojován od napájecího napětí celý pomocí P-kanálového MOS-FET tranzistoru. Schéma zapojení měniče, včetně zpětné vazby na nastavení výstupní úrovně 3,3V ukazuje Obrázek 5.6 .



Obrázek 5.6 Zapojení měniče ST L6920

5.1.5 Ostatní obvody

Jednotka je samozřejmě osazena ještě dalšími obvody nutnými pro její celkovou funkčnost či možnosti rozšíření jako např. vlastní stabilizátor napětí pro možnost napájení ze externího zdroje. Zapojení těchto součástek je standardní, a proto zde nejsou uvedeny.

5.2 Návrh zapojení převodníku USB-Nordic rfIRC35

Na tento převodník nebyly kladeny žádné velké nároky, proto byla jednotka optimalizována na co nejnižší cenu

5.2.1 Volba součástek pro převodník USB-Nordic

Mikrokontrolér MSP430 F147

Tento mikrokontrolér je podobný typu F1612, avšak je téměř o polovinu levnější, jelikož u této jednotky nejsou velké požadavky na velikost RAM či flash paměti. Zůstává zde ovšem hardwarová podpora SPI sběrnice a navíc má obvod UART, který je potřeba pro komunikaci s USB převodníkem, vyveden, na samostatných pinech, a není tak potřeba řešit oddělení od SPI.

Sériová paměť AT45DB011B

V požadavcích pro tento převodník byla možnost užití jako zprostředkovatele událostí, jako takový by musel být v tomto režimu téměř neustále na příjmu, nabízí se zde tedy možnost využít jej i jako tzv. „Logger“ síť, který by mohl zaznamenávat poslední události v bezdrátové síti. Pro tyto účely byla vybrána sériová flash paměť o kapacitě až 4Mbit.

Převodník USB-UART FT232R

Aby bylo možné jednoduše propojit jednotku postavenou na mikroprocesoru, který nemá hardwarově implementováno USB rozhraní, bylo třeba vybrat vhodný převodník. Tyto převodníky vyrábí firma FTDI která dodává i konfigurační software včetně ovladačů pro běžně užívané operační systémy. Tento převodník obsahuje navíc i integrovanou EEPROM paměť, do které je možné vložit identifikační číslo celého zařízení se kterým se pak bude hlásit pro operační systém. FT232R obsahuje i stabilizátor napětí a převodník úrovní. Je tedy možné propojit 3,3V úrovně s 5V úrovněmi a navíc až do odběru cca 100mA napájet zařízení umístěné za tímto převodníkem.

Převodník úrovní MAX3221

Z požadavku na zachování sériové linky RS232 bylo nutné propojit 3,3V úrovně procesoru MSP430 s 5V úrovněmi UART obvodu v PC. Pro tyto účely je velice vhodný obvod od firmy Maxim.

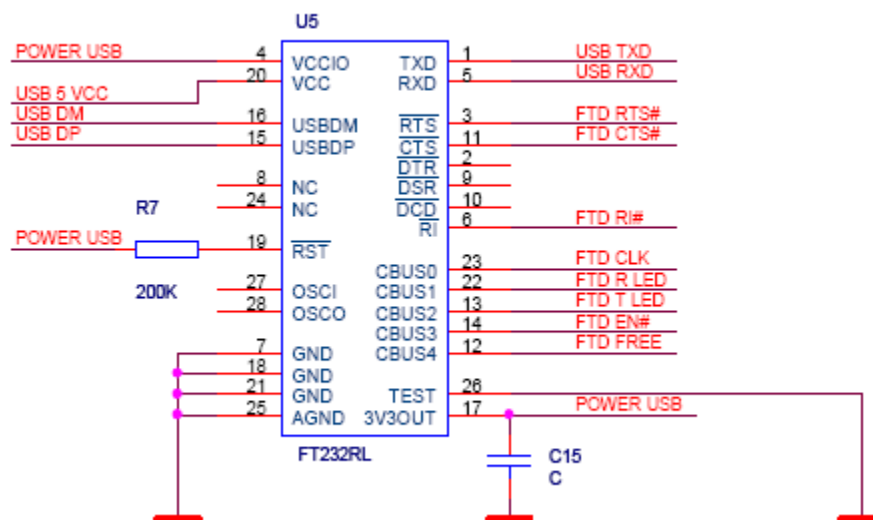
Bezdrátový obvod nrf905

Tento obvod je stejný jako u bezdrátového termostatu (viz. jelikož se jedná v podstatě o fyzickou vrstvu dané sítě.

5.2.2 Zapojení obvodu FTDI FT232R

Tento obvod umožňuje několik režimů práce a ze strany mikrokontroléru se jeví jako klasický UART, ze strany PC je k tomuto obvodu možno přistupovat pomocí ovladačů, které jej zobrazí jako virtuální COM port, či za pomoci DLL knihoven a přistupovat přímo jako k zařízení USB.

Schéma zapojení obvodu FT232R ukazuje Obrázek 5.7. U převodníku USB-Nordic je využito napájení přímo pomocí USB s úrovní 3,3V, to je ve schématu provedeno přivedením zpětné vazby z výstupu 3V2OUT na vstup VCCIO, který určí výstupní napěťové úrovně pro signály používané mikrokontrolérem MSP430. Mikroprocesor je připojen standardně, a jsou k němu navíc přivedeny řídicí signály pro řízení sériového přenosu



Obrázek 5.7 Zapojení obvodu FT232R

5.2.3 Ostatní obvody

Převodník USB-Nordic obsahuje i obvody, jejichž zapojení bylo vysvětleno již u bezdrátového termostatu a zde je zapojení stejné nebo velice podobné.

5.3 Návrh zapojení malého akčního členu rfIRC10

Při návrhu bylo nutné dodržet hlavně požadavek na veliký spínaný výkon a možnost napájení přímo ze síťového napětí 230V AC. Tento akční člen lze rozdělit v podstatě na 3 části:

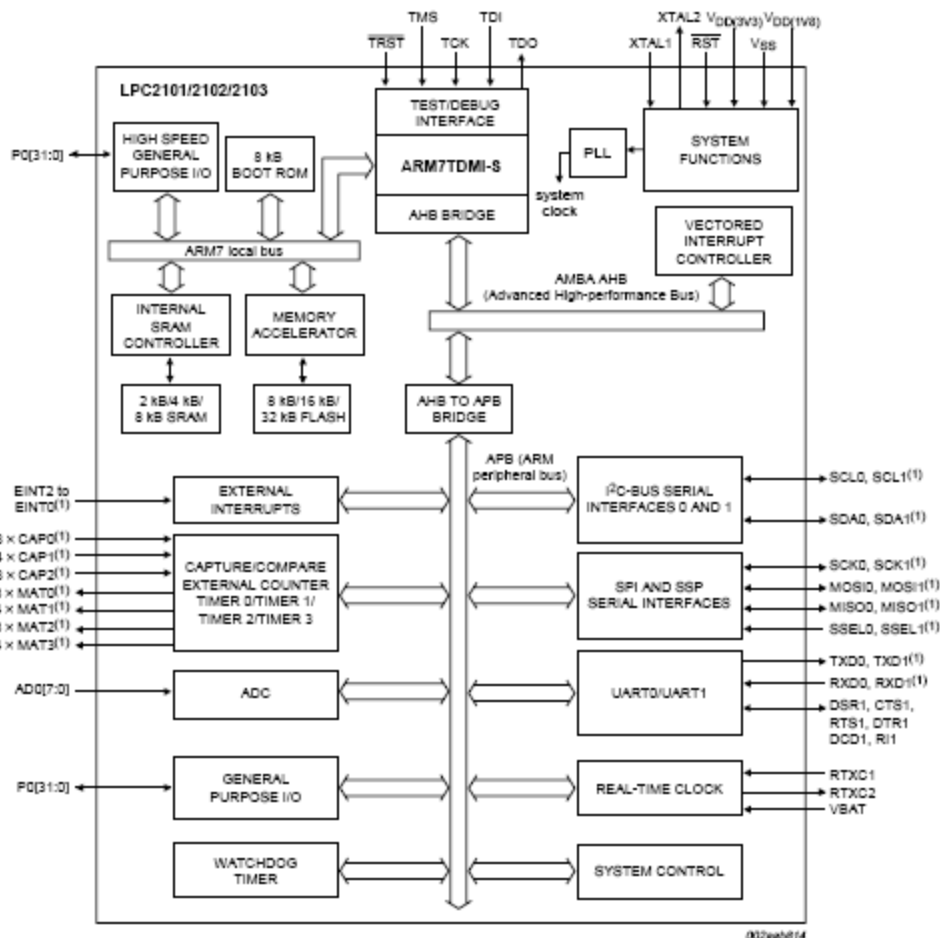
- Silová spínací část -výkonové spínací relé
- Zdrojová část - pulzní napájecí zdroj 230V ACDC - 5,0V DC
- Logická část - procesor a bezdrátový obvod

5.3.1 Volba součástek pro akční člen rfIRC10

Jelikož se jedná o zařízení, kterých v daném IRC systému může být poměrně hodně byl zde tlak na výslednou cenu zařízení ještě vyšší. Dále je zde trvalé napájení, takže není potřeba řešit extrémně nízkou spotřebu.

Mikroprocesor LPC2103

Tato jednotka systému je vybavená odlišným mikrokontrolérem RISC architektury ARM7. Jedná se o řadu 32 bitových procesorů LPC konkrétně LPC2103. Procesor obsahuje 8kB RAM a 32kB flash paměti. Podporuje SPI sběrnici důležitou pro komunikaci s bezdrátovým obvodem nrf905. Procesor má vestavěný bootloader umožňující programování pomocí sériové linky. Blokové schéma procesoru ukazuje Obrázek 5.8.



Obrázek 5.8 Blokové schéma mikrokontroléru LPC2103

5.3.2 Zapojení napájecího zdroje

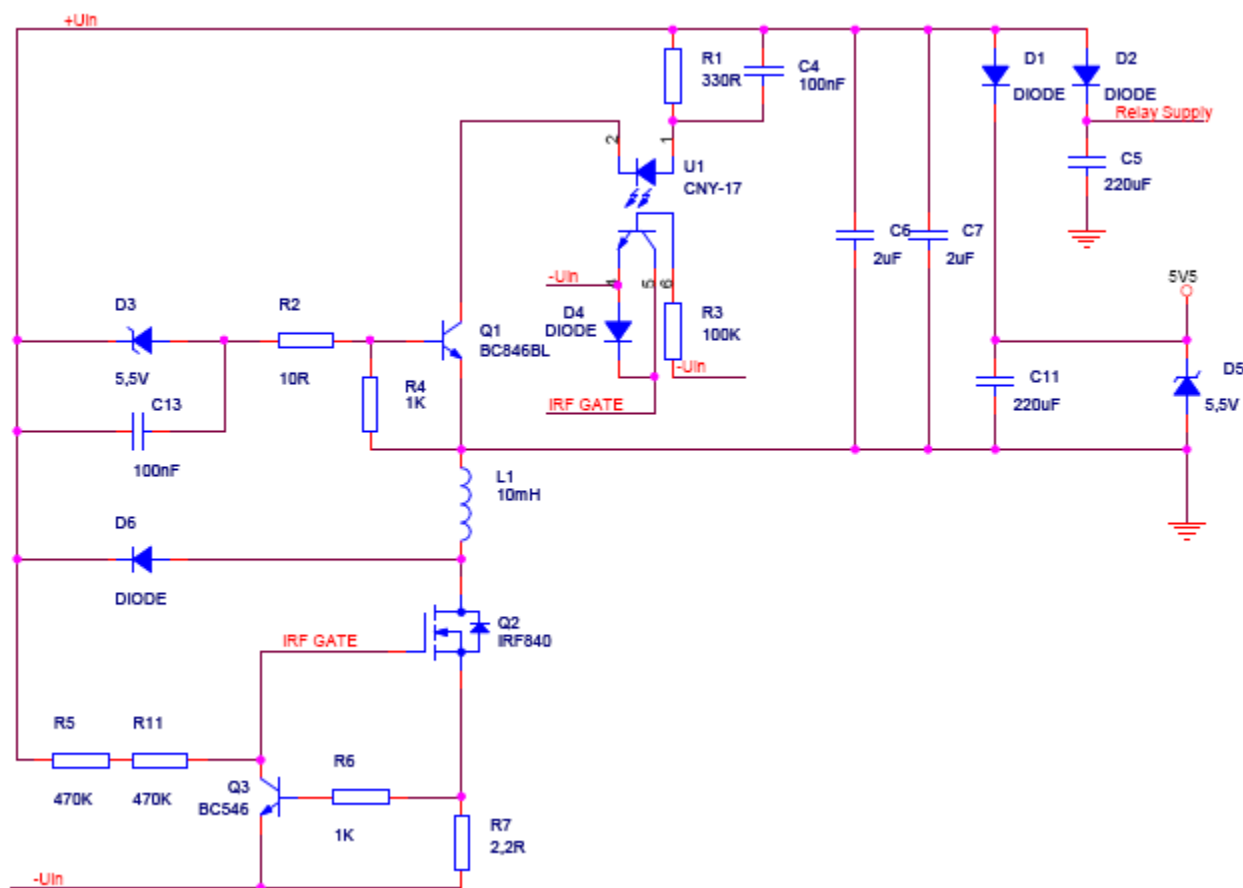
Jak již bylo zmíněno jednou z částí tohoto akčního členu je napájecí zdroj, který umožňuje napájení celého zařízení přímo ze síťového napětí 230V AC. Pro tuto aplikaci se nabízejí celkem 3 možnosti realizace napájecího zdroje:

1. zdroj s transformátorem, nevýhoda však je malá účinnost a s tím spojené zahřívání a zároveň i nekompaktní rozměry.
2. použití kapacitního děliče. I toto řešení má však nedostatky a to hlavně se zatěžovací charakteristikou, kdy je žádoucí aby zařízení odebíralo trvale

konstantní proud, což se v případě použití spínacího relé obtížně realizuje a vede též k vyšší spotřebě energie.

3. pulzní zdroj typu step-down, je konstrukčně nejsložitější, avšak umožňuje použití menších diskrétních součástek, a tím značně zmenšit konstrukci. Problém tohoto řešení je tak pouze odrušení nežádoucích harmonických složek které by mohly pronikat do sítě.

Na základě provedených rešerší bylo tedy navrženo schéma pulzního napájecího zdroje a provedena simulace v programu AD Multisim. Schéma napájecího ukazuje Obrázek 5.9. Tento zdroj byl navržen samostatně a je možné jej využít i k dalším aplikacím jako například již zmíněné parazitní napájení termostatu ze spínače osvětlení umístěného vedle. Zde by pak byla nutnost vyřešit galvanické oddělení.



Obrázek 5.9 Zapojení Step-Down konvertoru 230V ACDC/5,5V DC

Základem obvodu je komparátor Q1, jehož komparovací úroveň je pomocí zenerovy diody nastavena na 5,5V. Tím je v podstatě zaručena úroveň výstupního napětí. Tímto komparátorem je přes oddělovací optočlen U1 ovládán Gate použitého výkonového spínacího tranzistoru Q2, který akumuluje energii v indukčnosti L2. Při uzavření tranzistoru Q2 je proud cívku uzavírán přes diody D6, D1 a zenerovou diodou D5 taktéž 5,5V, která je paralelně připojena k akumulacím kondenzátorům C6 a C7 na výstupu tohoto obvodu. Zdroj obsahuje ještě ochranné obvody důležité pro práci s vysokými napětími.

Výstupní napětí na tomto zdroji jsou 2, jedno na zenerově diodě pro napájení logické části akčního členu, druhé přímo na filtračních kondenzátorech použité pro spínání výkonového relé.

Tomuto zdroji je ještě předřazen dvoucestný usměrňovač s filtračními a blokovacími kondenzátory.

V logické části akčního členu je mikroprocesor s bezdrátovým obvodem, zvláštnost tohoto mikroprocesoru je v použití dvou napájecích napětí (3,3V a 1,8V) proto je potřeba napětí zdroje pomocí stabilizátorů snížit na potřebné hodnoty.

5.4 Návrh zapojení velkého akčního členu rfIRC15

Návrh tohoto akčního členu zpracoval Zdeněk Tábořský v rámci projektu CEPOT s názvem „Sběr dat ze senzorů fyzikálních veličin s možností bezdrátové komunikace“. Toto zařízení do této diplomové práce přímo zapadá, a podle zadání bylo převzato.

5.4.1 Popis zapojení

Základem je mikrokontrolér MSP430 F167, který je pomocí sběrnice SPI připojen k bezdrátovému obvodu nordic nrf905. Jednotka je osazena 4 čtyřmi výkonovými relé, čtyřmi digitálními vstupy a jedním analogovým vstupem. Při modifikaci některých

obvodových prvků u obvodů digitálních vstupů je možné použít jeden z digitálních vstupů jako digitální výstup pro komunikaci se zdrojem tepla (kotle) např. pomocí protokolu OpenTherm.

Podrobný popis celého zapojení včetně schémat je v příložených souborech dokumentace k projektu CEPOT.

5.5 Návrh zapojení programátoru pro procesory LPC21xx

Procesory ARM řady LPC21xx od firmy Philips podporují mimo jiné i programování prostřednictvím sériového kanálu mezi PC a procesorem. Programování takovýmto způsobem má v sobě několik úskalí, a proto byla snaha vytvořit programátor, který umožní programování i u PC bez vyvedeného portu RS232 a zároveň odstraní nedostatky programování jen s použitím obyčejného sériového kabelu, které jsou spojené s různými režimy startu procesorů LPC.

5.5.1 Popis zapojení

Hlavní částí programátoru je převodník USB/UART FT232R, u kterého je pro programování využito ovladačů pro virtuální COM port a lze tak používat standardní softwarové nástroje, jako by se programovalo přímo pomocí linky RS232. Tento obvod byl přesněji popsán v na straně 22 v popisu zapojení převodníku USB-Nordic.

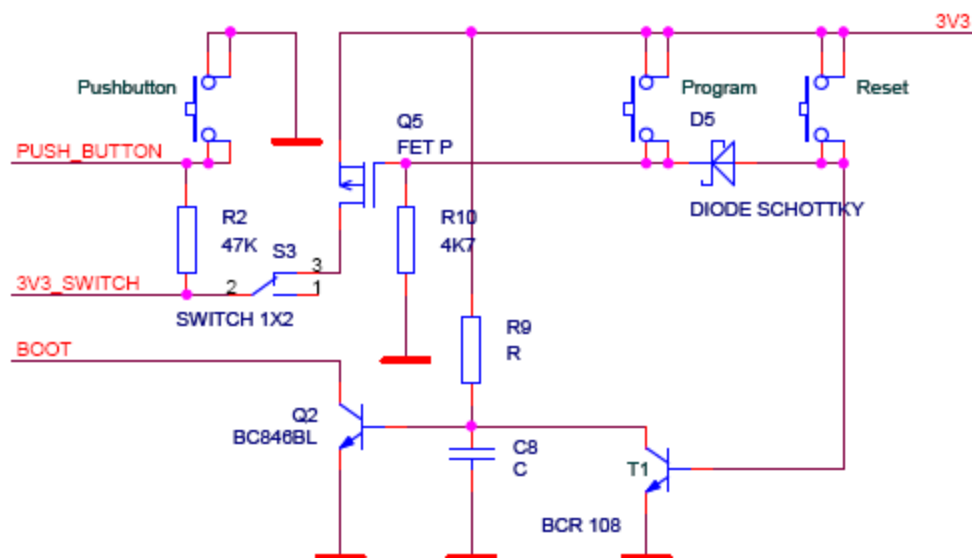
Další částí je napájení programovaného zařízení přímo z PC. Programátor je tedy vybaven vlastním stabilizátorem napětí na úroveň 3,3V potřebnou pro napájení procesorů LPC. Pro případ vlastních stabilizátorů na programovaném zařízení, je na programovacím konektoru vyveden přímo 5V výstup ze sběrnice USB. Pro programování aplikací, které mají spotřebu menší jak 500mA není tedy nutné používat externí zdroj napětí.

Pro správné naprogramování procesorů LPC21xx je nutné zajistit správnou sekvenci mezi signály RESET, BOOT, a připojením napájecího napětí. Pomocí různých časových sledů procesor přejde buď do programovacího režimu, nebo do režimu běhu programu. Správný sled těchto signálů je vyřešen pomocí logiky ovládané tlačítka RESET a PROGRAM. Schéma této logiky ukazuje Obrázek 5.10.

Při stisku tlačítka PROGRAM, dojde k odpojení výstupu 3V3_SWITCH od napájecího napětí a tím je možné ovládat tranzistor, který přitáhne RESET procesoru na logickou úroveň „LO“. Signál BOOT zůstává v tomto případě díky tranzistoru Q2 přitážen taktéž k úrovni „LO“ a puštěním tlačítka program přejde procesor do programovacího režimu. Stiskem tlačítka RESET dojde opět k odpojení výstupu 3V3_SWITCH, ale zároveň i k odpojení výstupu BOOT, který musí být na straně procesoru tažen pomocí PULL-UP rezistoru tažen k úrovni „HI“. Po uvolnění tlačítka dojde k opětovnému připojení výstupu 3V3_SWITCH k napájecímu napětí a tím uvolnění RESET signálu procesoru, ovšem přitážení signálu BOOT je nyní zpožděno díky nabíjení kondenzátoru C přes odpor R, pomocí nichž je možné nastavit časovou konstantu. V tomto režimu je procesor připraven k běžné práci s naprogramovaným kódem.

Pro programování a ladění zařízení, která nejsou osazena vstupními tlačítky, je programátor osazen vlastním tlačítkem které je možné připojit na libovolný volný vstupní signál procesoru a tak řídit události uvnitř laděného programu.

Programátor je též vybaven několika LED diodami indikujícími aktuální stav a činnost programátoru.



Obrázek 5.10 Zapojení části pro obsluhu boot režimů procesoru LPC2103

Kapitola 6

Návrh desek plošných spojů

Tato kapitola se zabývá přesným řešením desek plošných spojů až na úroveň zástavby jednotlivých modulů do použitých krabiček definované velikosti a tvaru. Pro návrh bylo využito prostředí OrCAD Layout v návaznosti na OrCAD Capture CIS.



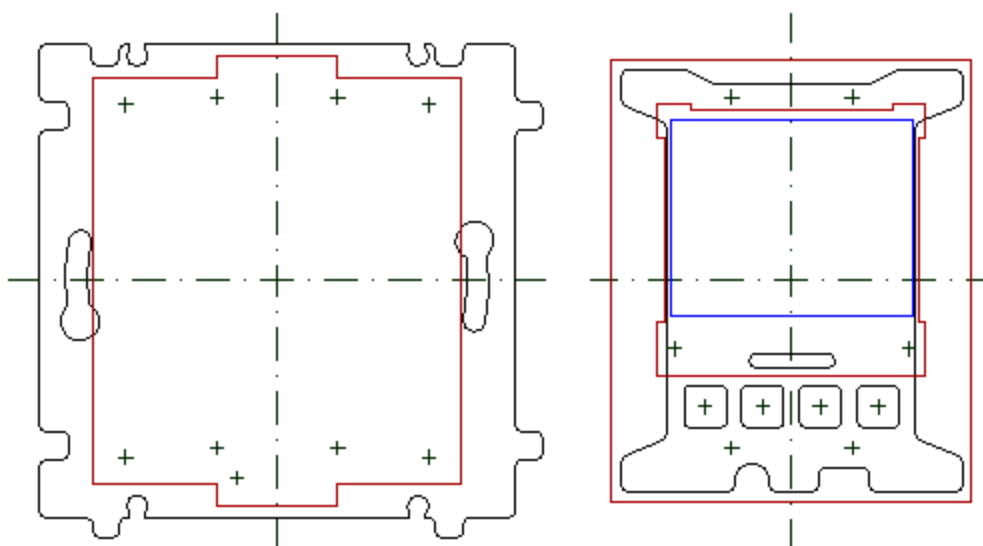
Obrázek 6.1 Vzhled jednotky termostatu rfIRC25

6.1 Návrh DPS pro bezdrátový termostat rfIRC25

Vzhledem k nutnosti zástavby do krabičky zde bylo nutno řešit jak prostorové uspořádání jednotlivých komponent do použité krabičky, tak vlastní tvar desky, která by měla s krabičkou ABB Time lehce smontovat.

6.1.1 Návrh obrysů desek pro ABB Time v prostředí AutoCAD

Jelikož se jedná o zařízení u něhož se předpokládá komerční využití je proto nutnost vyřešit tuto část tak, aby nic nebránilo jednoduché výrobě a případné obsluze ze strany uživatele. Aby bylo možné toto provést, musí notně jedna z použitých desek tvořit spodní část celé krabičky, z tohoto důvodu byla vzata inspirace, přímo z konstrukce krabičky a obrys spodní desky je přesně navržen namísto původní plastové části. Jelikož se jedná o prostorové uspořádání, bylo nutné řešit ještě desku, která ponese displej a tlačítka. Tato deska musí přesně zapadat do vrchního krytu ABB Time, zároveň je ale nutné aby se tato deska vyhýbala bateriím a nezvedala celkový profil krabičky. Z tohoto požadavku vznikl tvar, který je v takové výšce, že umožní přidržení baterií na správném místě a zároveň umístí displej s tlačítky co nejnižše aby zůstal zachován profil krabičky. Náhled obrysů obou desek včetně důležitých omezení a bodů ukazuje Obrázek 6.2. Tyto obrysy byly následně importovány do vývojového prostředí OrCAD a tvoří speciální vrstvu, pomocí níž výrobce vyfrézuje potřebný tvar DPS.



Obrázek 6.2 Obrisy horní a dolní desky

6.1.2 Prostorové uspořádání DPS a komponent na DPS

Obě desky jsou umístěny nad sebou a jsou propojeny pomocí pinových lišt s roztečí 2mm a počtem 10 pinů na spodní i vrchní části. Důležitou částí celého návrhu je uchycení baterií. Byly použity bateriové klipy od firmy Keystone primárně určené pro u nás ne zcela běžné baterie s označením AAAA. Jsou použity nikoliv klasicky, ale jsou pootočený o 90° čímž umožní jak kontakt pro baterie, tak na druhé straně přichycení rámečku ABB Time. Takto je maximálně usnadněna montáž termostatu do standardní přístrojové krabice a tato konstrukce umožňuje zlevnění díky odpadajícím operacím, kterými by bylo například lepení či šroubování.

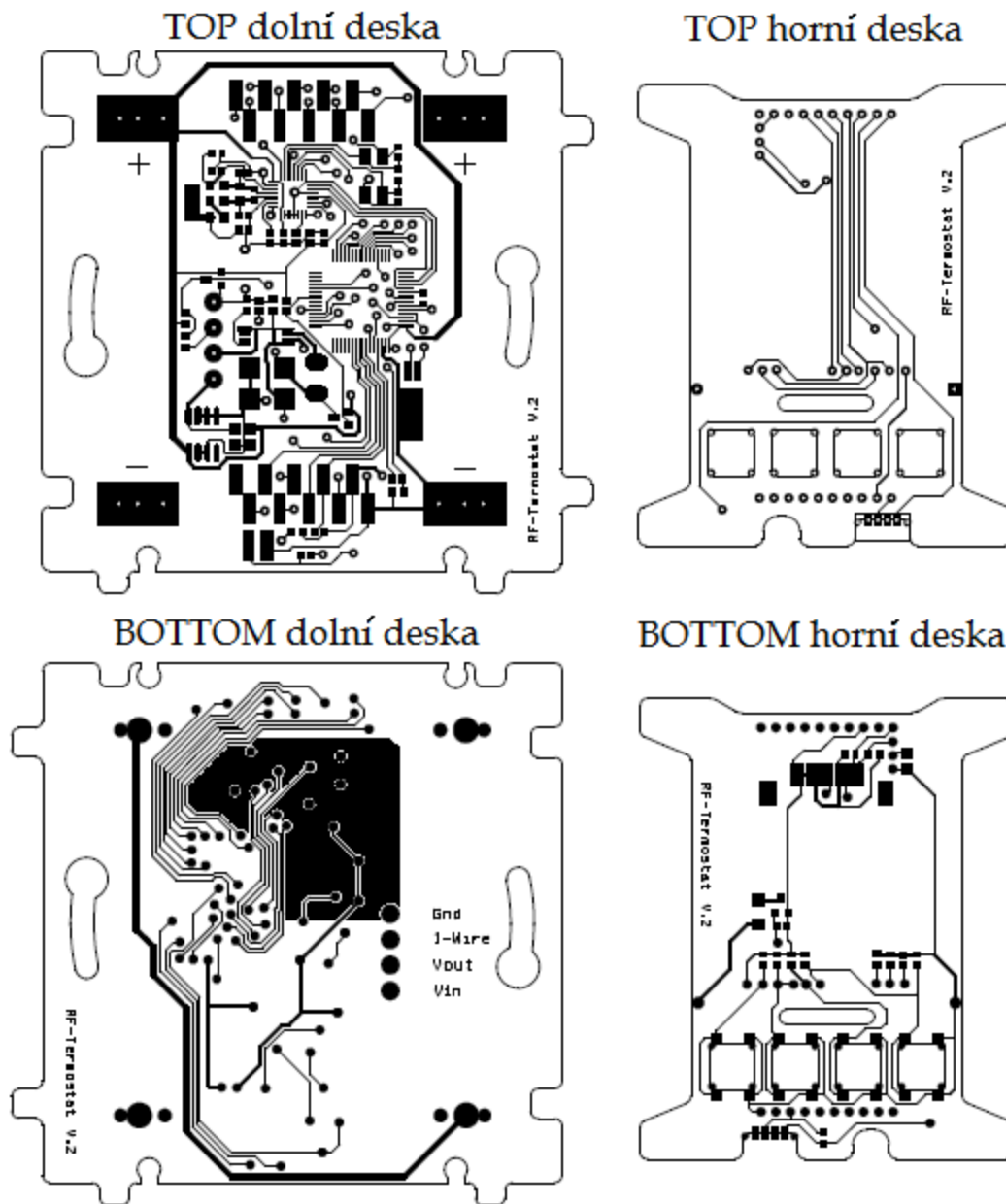
Při umístění displeje bylo použito originálního podsvětlovacího rámečku, který umožnil přesné umístění displeje na horní desce. Poslední mechanickou částí DPS jsou tlačítka. Protože je jednotka navržena tak, aby bylo použito co nejméně částí, které by vyžadovaly nějakou modifikaci či dokonce vlastní výrobu. Byla snaha osadit ji tlačítka, které mají vlastní bílé hmatníky, které zapadnou přímo do designu ABB. Problém však nastal s příliš velkou výškou tlačítek, byla proto zvolena netypická koncepce použití SMD

tlačítek, ale s osazením skrz DPS. Proto byly přidány do obrysů pro frézování ještě díry na prostrčení SMD tlačítek. Tato konstrukce nejen umožnila snížení tlačítek, ale i zpřesnila jejich usazení.

Při rozmisťování součástek na DPS, byla snaha umístit všechny obvody na spodní desce. Toto umožňuje variabilitu a použití bezdrátového termostatu jen jako levného čidla teploty v případě neosazení tlačítka displejem. Dále bylo řešeno rozmístění čidel teploty a vlhkosti. Tato čidla byla umístěna do spodní části, kde se předpokládá nejmenší vliv tepelných ztrát jednotlivých komponent.

Anténní část byla zvolena pro prutovou čtvrtvlnnou anténu s desimetrizačním členem. Nebyla volena varianta leptání prutu přímo na DPS, jelikož při frekvencích kolem 1GHz se již uplatňují parazitní parametry vodičů leptaných na DPS a proto je pro anténu připravena pouze ploška.

Obrázek 6.3 ukazuje náhled DPS pro bezdrátový termostat rfiRC25. Návrh desek plošných spojů je součástí příloženého CD.



Obrázek 6.3 Náhled DPS bezdrátového termostatu rfIRC25

6.2 Návrh DPS pro převodník USB-Nordic rfIRC35

U této jednotky byl pouze požadavek na použité rozhraní USB a RS232. Vybraná krabička od firmy Gainta má rozměry 90x50x24mm je vyrobena z šedého ABS plastu .

6.2.1 Návrh obrysu desky podle použité krabičky

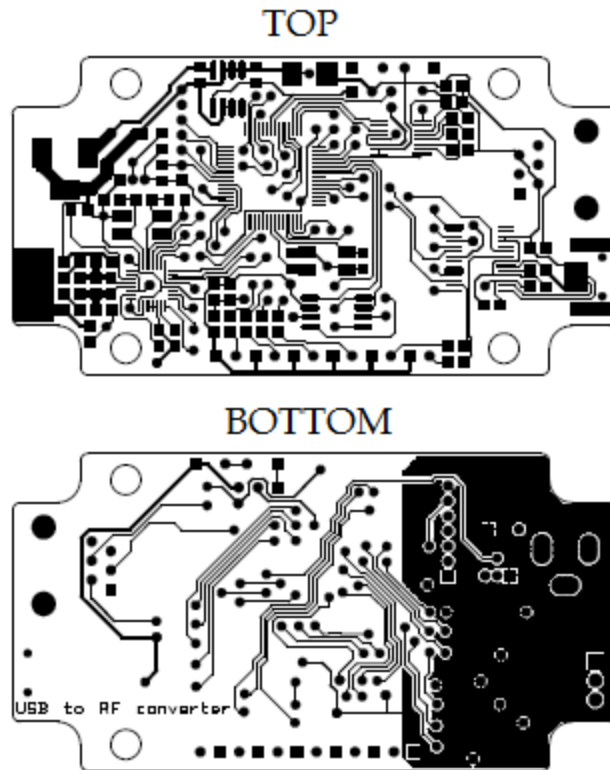
Přesné usazení DPS do krabičky umožní přesné strojové obrábění čel krabičky pro použité konektory.



Obrázek 6.4 Převodník USB-Nordic rfIRC35

6.2.2 Prostorové uspořádání komponent na DPS

Při návrhu DPS byl brán ohled hlavně na použitou krabičku, a od toho se odvíjely i požadavky na co nejmenší pouzdra součástek. Pro komunikaci pomocí RS232 je standardem konektor Canon DB9, ten však zabírá poměrně hodně místa proto bylo zvoleno mírně nestandardní řešení pomocí konektoru RJ11. Toto řešení využívají například některá PLC. Pro USB komunikaci je vybrán mini USB konektor používaný v celé řadě zařízení. Připojení externí antény je realizováno pomocí závitů M4, pro který je možné zakoupit čtvrtvlnnou prutovou anténu. Ukazuje náhled DPS převodníku USB-nordic. Návrh DPS a rozpiska součástek je součástí přílohy



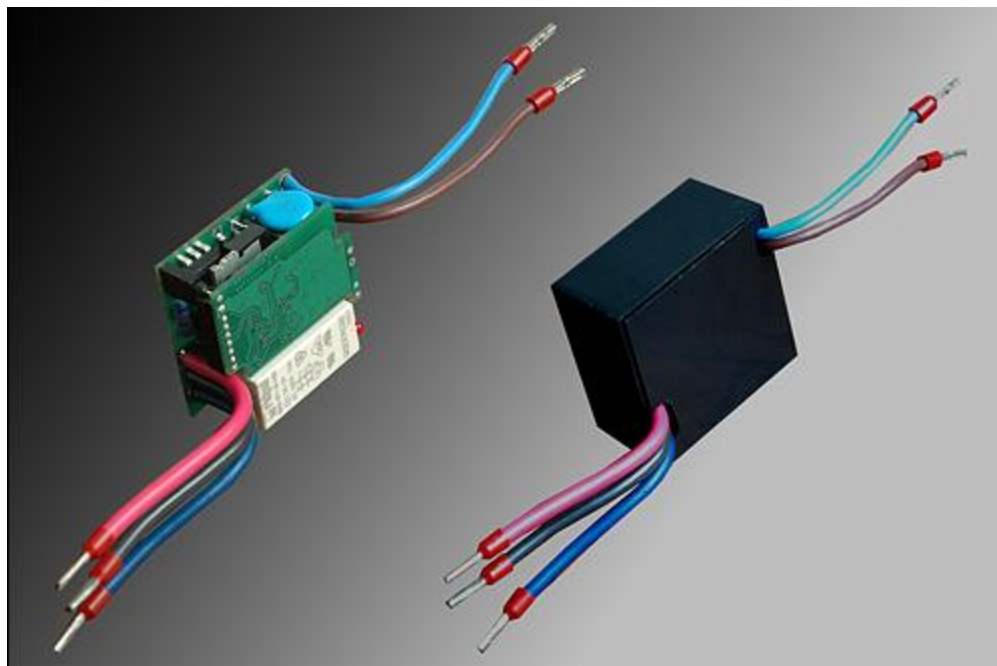
Obrázek 6.5 Náhled DPS převodníku rfIRC35

6.3 Návrh DPS pro malý akční člen rfIRC10

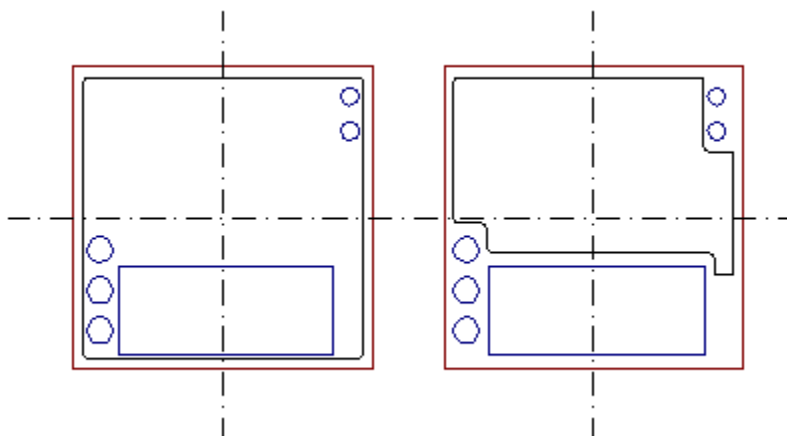
Při návrhu desek plošných spojů byl kladen důraz na velikost celého zařízení, tomu odpovídala i použitá pouzdra jednotlivých součástek. Zařízení je složeno ze dvou desek.

6.3.1 Návrh obrysů desek podle použité krabičky

Návrh se soustředil hlavně na horní desku, která se díky použitému výkonovému relé nevešla nad celou spodní desku, dále bylo potřeba vynechat místo pro napájecí a silové vodiče, které jsou vedeny od spodní desky. Obrázek 6.7 ukazuje návrh obrysů včetně omezení (relé a vodičů)



Obrázek 6.6 Akční člen rfIRC10



Obrázek 6.7 Návrh obrysů malého akčního členu

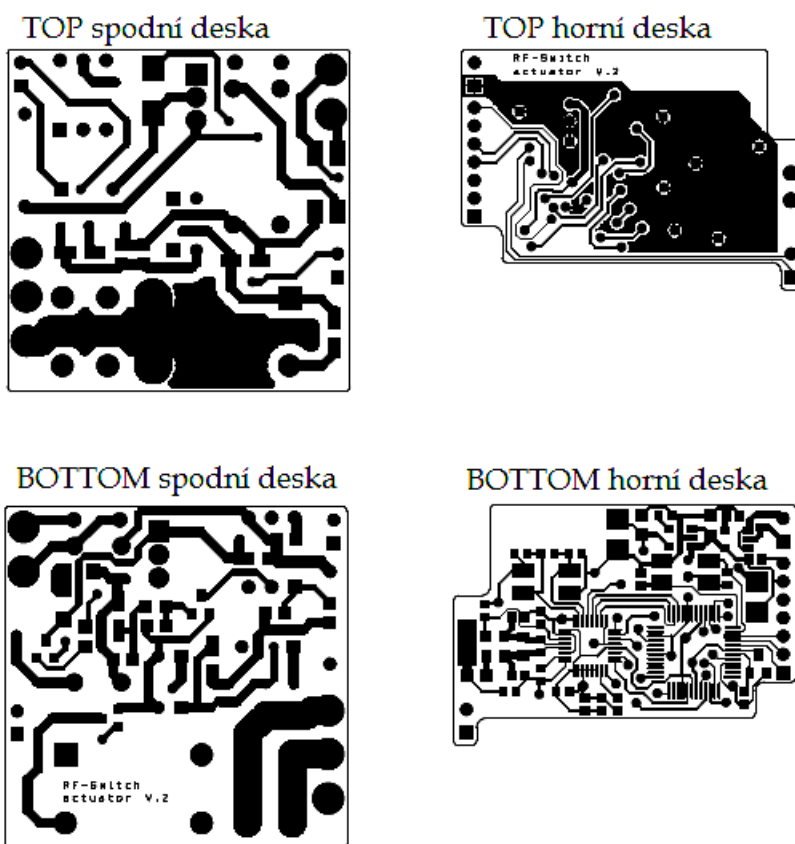
6.3.2 Prostorové uspořádání DPS a komponent na DPS

Spodní deska je tvořena zapojením napájecího zdroje a silové části. Zde byl brán ohled hlavně na výkonovou ztrátu, průrazné napětí a co možná nejmenší rozměry použitých pouzder. Při návrhu DPS byl brán ohled na izolační vzdálenosti mezi jednotlivými vodiči.

Silová část spodní desky je tvořena výkonovým relé a zakončením pro připájení silových vodičů se jmenovitým průřezem 1,5mm². Silové spoje k relé jsou provedeny s pájivou maskou, a budou následně pocínována pro zvýšení jejich průřezu a tím snížení tepelné ztráty.

Horní deska slouží pro ovládání relé a komunikaci s ostatními komponentami systému. Zde jsou tedy umístěny stabilizátory napětí pro 3,3V a 1,8V procesor a bezdrátový obvod. Deska je kvůli tloušťce osazena pouze z jedné strany a je zde využito převážně pouzder SMD 0603.

Spojení obou desek tvoří dutinkové lišty na spodní desce a pinové lišty na horní desce. Tyto lišty jsou na desce umístěny diagonálně a zaručují tak mechanickou pevnost a přesné usazení desek proti sobě. Obrázek 6.8 ukazuje náhled výsledných DPS. Návrh desek je součástí příloženého CD



Obrázek 6.8 Náhled DPS malého akčního členu rfIRC10

6.4 Návrh DPS pro velký akční člen rfIRC15

Návrh DPS tohoto akčního členu zpracoval Zdeněk Tábořský v rámci projektu CEPOT s názvem „Sběr dat ze senzorů fyzikálních veličin s možností bezdrátové komunikace“.

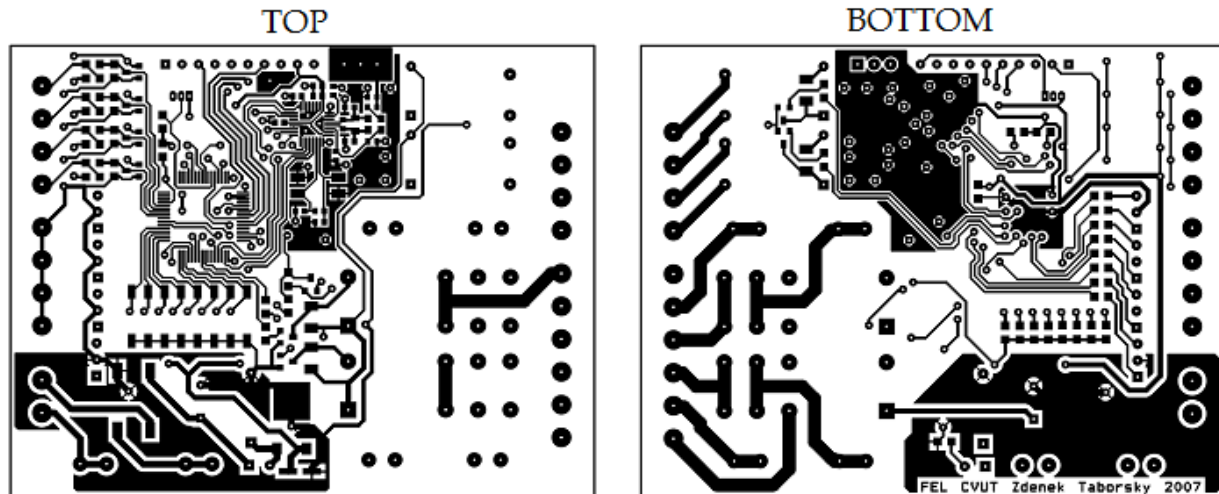
6.4.1 Stručný popis návrhu DPS pro akční člen rfIRC15

DPS byla navrhována s ohledem na použitou krabičku **krabička** s rozměry 120 x 80 x 41mm z ABS plastu, ve světle šedé barvě (RAL 7035), s možností upevnění na stěnu. Označení krabičky je **ABS75-GRY** a její design ukazuje Obrázek 6.9.



Obrázek 6.9 Krabička pro vestavbu akčního členu rfIRC15

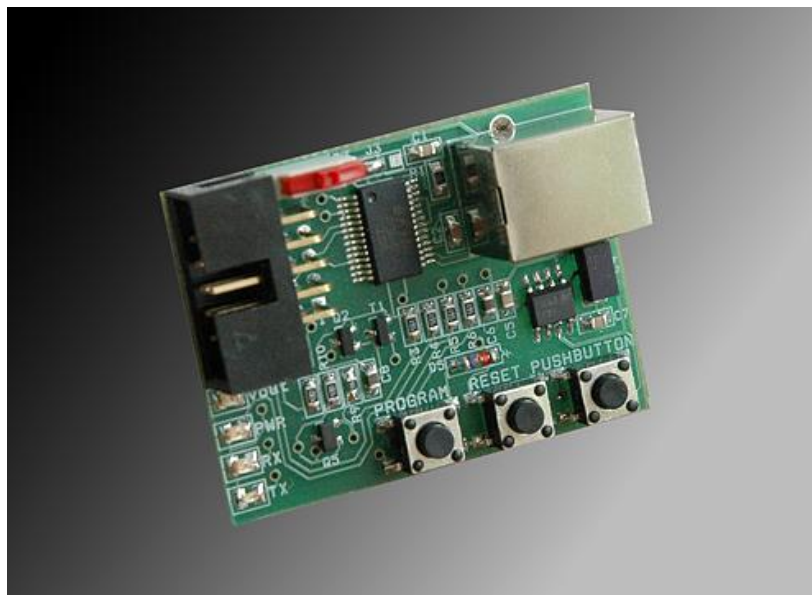
Při návrhu bylo dále řešeno striktní oddělení silové a logické části, proto byla namísto původně zamýšlených výkonových MOSFET tranzistorů zvolena výkonová relé, která řeší galvanické oddělení. Obrázek 6.10 ukazuje výslednou DPS pro akční člen rfIRC15.



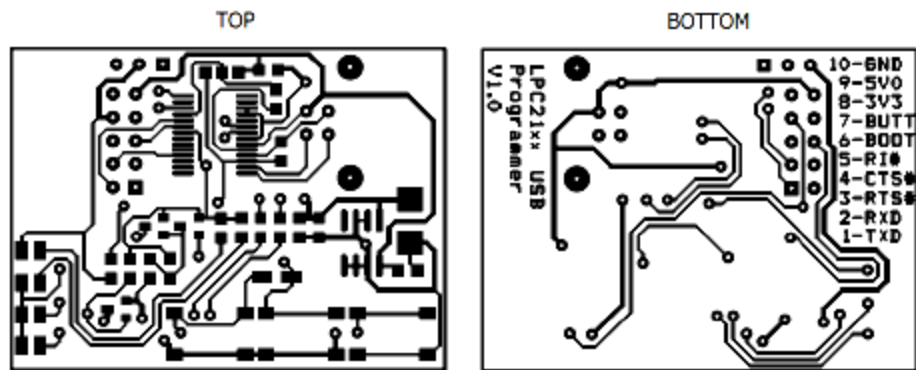
Obrázek 6.10 Náhled DPS pro velký akční člen rfIRC15

6.5 Návrh DPS pro programátor procesorů LPC21xx

Pro programátor LPC byl zvolen USB konektor typu B, pro výstup klasický desetipinový konektor umožňující připojení plochého vodiče. U tohoto zařízení nebyla volena zástavba do krabičky, jelikož se jedná pouze o výpomocný obvod. Obrázek 6.12 ukazuje náhled DPS, celkový návrh je součástí přiloženého CD.



Obrázek 6.11 USB programátor procesorů LPC21xx



Obrázek 6.12 Náhled DPS USB programátoru pro LPC21xx

Kapitola 7

Implementace základního software

Při vývoji software bylo využito možností prostředí MspGCC, což je mezičlánek pro použití jazyka GNU C pro mikrokontroléry Texas Instruments řady MSP430. Pro vývoj základního softwaru pro malý akční člen rFIRC10, osazeného mikrokontrolérem LPC2103, bylo použito prostředí WinARM, což je obdoba MspGCC, podporující architekturu ARM.

7.1 Popis knihoven pro obsluhu hardware

Zde budou probrány knihovny, jejichž činnost je přímo vázaná na použitý procesor.

7.1.1 Knihovna pro obsluhu čidla DS18B20 (ds18b20.c, ds18b20.h)

Jde v podstatě o softwarovou implementaci knihovny OneWire sběrnice pro použití tohoto externího senzoru

Ds18b20ResetPulse ()

Funkce pro inicializaci OneWire rozhraní. Komunikaci zahajuje master(mikroprocesor) vysláním reset pulzu o délce 480us.

Ds18b20PresencePulse ()

Funkce čekání na prescense pulz. Po odeslání reset pulzu čeká master na odpověď od čidla teploty v podobě prescense pulzu.

Ds18b20Write1 ()

Funkce pro odeslání bitu úrovně „1“ na sběrnici OneWire. Tato funkce je realizována bez použití časovačů, jelikož se jedná o velice rychlou komunikaci v řádech jednotek mikrosekund.

Ds18b20Write0 ()

Funkce pro odeslání bitu úrovně „0“ na sběrnici OneWire. Princip funkce je stejný jako u předchozí.

Ds18b20Read ()

Funkce pro přijetí bitu ze sběrnice OneWire. Tato funkce je taktéž realizována bez použití časovačů. Vrací hodnotu přijatého bitu.

Ds18b20WriteByte (byte command)

Funkce slouží k odeslání bytu na sběrnici OneWire. Tato funkce podle vloženého parametru *command* odešle byte pomocí volání funkcí *Ds18b20Write0()* a *Ds18b20Write1()*.

Ds18b20SkipRom ()

Tato funkce slouží k přeskočení algoritmu ohlašování čidel rozsáhlé sběrnice a dojde k adresaci jednoho čidla. Volá funkci *Ds18b20WriteByte ()* s parametrem *0xCC*.

Ds18b20ConvertT ()

Funkce odesílá na sběrnici příkaz pro převod teploty. Čidlo, které bylo zadresováno jej přijme a zahájí převod. Vrací hodnotu 1 pokud byl převod dokončen.

Ds18b20ReadScratchpad ()

Funkce vyše příkaz pro vyčtení hodnoty registrů, ve kterých je uložena informace o teplotě. Následně pomocí volání funkce *Ds18b20Read ()* načte tyto hodnoty do proměnné *scratch_pad[]*.

Ds18b20GetTemperature ()

Tato funkce využívá volání předešlých funkcí, pomocí nichž je možno realizovat odečet teploty ze senzoru DS18B20. Funkce nejprve resetuje rozhraní a dále čeká na ohlášení čidla. Jakmile se čidlo ohlásí prescense pulzem, dojde k adresaci jednoho teplotního čidla pomocí funkce *Ds18b20SkipRom()* a následně je odeslán příkaz pro převod teploty, který na sběrnici odešle funkce *Ds18b20ConvertT()*, která také následně oznámí ukončení převodu návratovou hodnotou 1. V tomto okamžiku je převedena teplota v čidle DS18B20 a je připravena k vyčtení. V tomto okamžiku dojde k zahájení nového přenosu a dojde k vyčtení teploty pomocí funkce *Ds18b20ReadScratchpad()*.

Formát dat s informací o teplotě

Informace o teplotě je uložena ve dvou bytech s rozlišením 0,0625°C, formát dat ukazuje Obrázek 7.1. Každý bit ukazuje zastoupení jisté mocniny dvou. Výsledná teplota je dána součtem mocnin jež jsou bitově zastoupeny v přijatých datech. Nejvyšších 5 bajtů (označeno písmenem S) z informace o přijaté teplotě ukazuje zda se jedná o teplotu kladnou či zápornou.

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2^5	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	2^6	2^5	2^4

Obrázek 7.1 Formát dat získaných z čidla teploty DS18B20

7.1.1 Knihovna pro interní čidlo teploty (*teplomer.c*, *teplomer.h*)

Procesory MSP430 řady F14x a vyšší mohou pomocí AD převodníku získat informaci o teplotě přímo z interního PN přechodu.

GetTemperature ()

Funkce která obsahuje rutiny inicializace AD převodníku, dále dojde k povolení interní napěťové reference s napětím 2,5V. Pomocí AD převodníku změří napětí na teplotním senzoru a převede jej na číslo odpovídající teplotě mikroprocesoru a nakonec odpojí napěťovou referenci a uvolní AD převodník. Jelikož je příkon procesorů MSP430 velice nízký, teplota díky malé výkonové ztrátě přímo odpovídá teplotě okolí.

Funkce vrací celé číslo, jež odpovídá desetinásobku skutečné teploty. Výpočet pro převod získaného čísla na teplotu je převzat z datasheetu [1] k procesorům řady F16x/F16xx.

7.1.2 Knihovna pro obsluhu displeje EM6125 + I2C (*LCD.c*, *LCD.h*)

Rozbor formátu dat pro zobrazení na displeji

Displej má velikost 102x81 bodů. Komunikace s displejem probíhá po sběrnici I2C, displej se ovládá sadou inicializačních příkazů. Po této inicializaci jsou posílána data do bank paměti RAM displeje. Každá banka obsahuje 102 sloupců o 8 bajtech. První řádek displeje je pro ikony, a zapisuje se do něj vždy pouze 1 bit z celého bajtu. Zdali se bude zapisovat MSB či LSB záleží na způsobu adresování nastaveného v inicializaci. Použitý displej obsahuje celkem 10 plnohodnotných bank.

Plocha displeje je realizována pomocí pole bytů zvaného *LcdCache[]* toto pole má velikost 102x10 bytů a zabírá tak značnou část paměti RAM. Byla snaha o vykreslování displeje s polem o poloviční velikosti (102x5 bytů), ale toto vedlo k nekonzistenci mezi daty zobrazenými na ploše LCD a daty v poli *LcdCache[]* . Implementace funkcí pro toto vykreslování zůstala zachována, avšak vzhledem k využití displeje pro kreslení složitější grafiky se tato koncepce nevyužívá.

I2CInit ()

tato funkce provede inicializaci a rozhraní I2C a následně resetuje displej, což je nutné pro jeho první start.

LcdBrightness (byte Brightness)

tato funkce modifikuje inicializační data tak, aby bylo možné dosáhnout jiné barevné teploty pixelů zobrazených na LCD. Lze nastavit hodnotu od v intervalu od -20 do +20.

LcdContrast (byte Contrast)

Funkce pro nastavení kontrastu displeje, modifikuje inicializační data pro změnu úrovně napětí displeje což se projeví jako změna kontrastu, lze nastavit hodnoty od 0 do 2.

LcdSleep ()

Funkce nastavuje do inicializačních dat flag pro převedení displeje do sleep módu. Nejprve dojde k odpojení plochy displeje od napětí parametrem *blank* a poté dojde k uspání, při němž klesá spotřeba displeje pod $1\mu A$.

LcdIcon (byte icon)

Funkce nastavuje, zdali bude řádek ikon zobrazen či ne, nastavení je pouze v inicializačních datech a hodnoty v poli *LcdCache[]* tento řádek neovlivní.

LcdGotoXY (byte x, byte y)

Funkce spočítá pozici kurzoru pro výpis textu podle zadaného řádku a sloupce. Řádky a sloupce jsou v tomto případě dány výškou resp. šířkou fontu. Horizontální polohu *x* lze tedy nastavit 1 až 17, vertikální polohu *y* lze nastavit 1 až 10.

LcdGotoXYE (byte x, byte y)

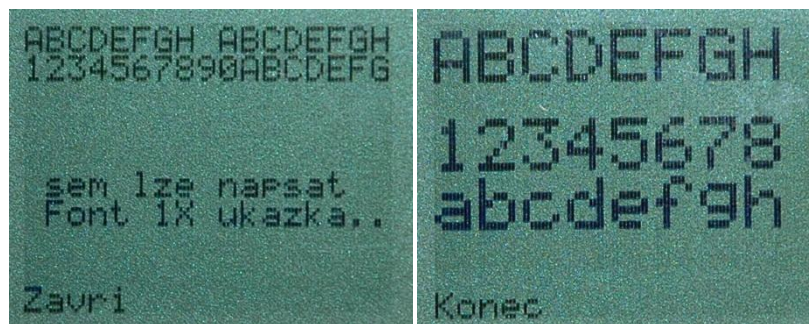
Funkce podobná předchozí, sloupce jsou ale dány jednotlivými pixely. Tato funkce je pro použití přesnějšího formátování textu. Horizontální polohu *x* lze tedy nastavit 0 až 101, vertikální polohu *y* lze nastavit v rozsahu 0 až 9.

LcdChar (byte size, byte ch)

tato funkce vybere podle vloženého znaku příslušný obraz z ASCII tabulky fontu 5x7 a vloží jej do pole *LcdCache[]*. Pro velikost fontu 2 je spočítán obraz horní a dolní poloviny každého sloupečku původního fontu, přičemž první je 2x horní polovina je zapsána do pole na pozici *LcdCache [index]* a *LcdCache[index+1]*, dolní polovina každého sloupečku původního fontu je zapsána na pozici *LcdCache[index+101]* a *LcdCache[index+102]*. Na začátku této funkce je nastaven vodoznak *LoWaterMark* na hodnotu *index*, pokud je tato hodnota menší jak *LoWaterMark*. Na konci této funkce je nastaven horní vodoznak *HiWaterMark* na hodnotu *index* při fontu velikosti 1 a na hodnotu *index+102* pro velikost fontu 2. Obrázek 7.2 ukazuje výsledné fonty zobrazené na LCD.

LcdStr (LcdFontSize size, byte *dataPtr)

Funkce pro výpis textu, volá funkci *LcdChar (byte size, byte ch)* a předává jí jako parametr velikost fontu a ukazatel na znak v řetězci.

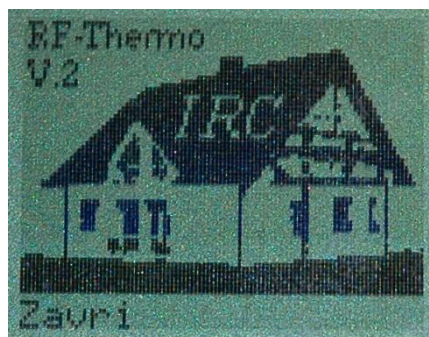


Obrázek 7.2 Náhled fontu velikosti 1 a 2

LcdPixel (byte x, byte y, byte mode)

tato funkce vloží podle zadaných parametrů pixel do pole *LcdCache[]*, pokud je proměnná *mode* rovna 0, pak je pixel smazán, pokud 1, pixel je vložen, pokud 2 pak se provede operace XOR s původním pixelem v poli *LcdCache[]*. *HiWaterMark* i *LoWaterMark* jsou rovny hodnotě *index* pokud je tato hodnota vyšší resp. nižší jak původní vodoznaky. Obrázek 7.3 ukazuje užití této funkce pro vykreslení bitmapy. Jde

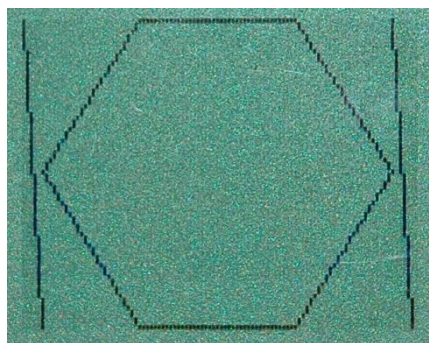
jen o demonstraci funkce vzhledem k počtu operací je vykreslování neformátovaného obrázku pomocí této funkce velice náročné na strojový čas.



Obrázek 7.3 Ukázka vykreslení obrázku pomocí funkce *LcdPixel* ()

LcdLine (byte x1, byte y1, byte x2, byte y2, byte mode)

Funkce pro kreslení obecné čáry tato funkce z vložených souřadnic [x1,y1] a [x2, y2]. Tato metoda spočítá přírůstek v ose x a v ose y a následně volá funkci *LcdPixel (byte x, byte y, byte mode)* které předává informaci o módu kreslení a spočítané souřadnice jednotlivých bodů přímky. ukazuje funkci pro kreslení obecných čar.



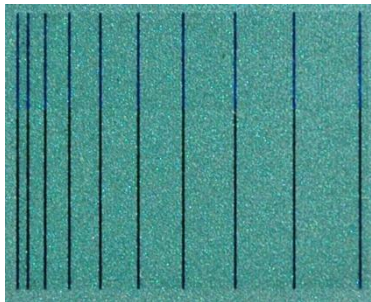
Obrázek 7.4 Náhled obrazce vytvořeného pomocí funkce *LcdLine* ()

LcdVLine (int xv ,int y1v, int y2v, int modev)

Funkce pro rychlé kreslení vertikálních čar. Jelikož je při kreslení obecných potřeba velké množství operací včetně volání funkce *LcdPixel (byte x, byte y, byte mode)* je toto vykreslování velice pomalé a při práci se složitější grafikou neúnosně zatěžuje procesor.

Vzhledem k tomu, že mnoho objektů používaných v grafickém prostředí je většinou složeno ze svislých a vodorovných čar, byly proto napsány funkce pro rychlé kreslení horizontálních a vertikálních čar. Tyto funkce pracují přímo s daty v *LcdCache* [].

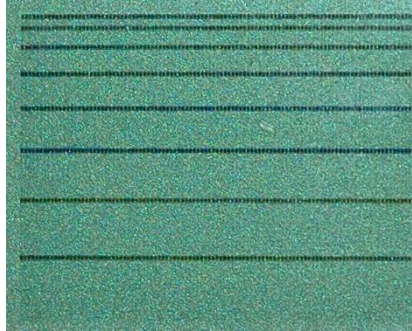
Princip funkce *LcdVLine()* je založen na předdefinovaných vertikálních maskách, které jsou podle nadefinovaných pravidel vkládány přímo do pole *LcdCache* []. Nejprve je proveden výpočet, zdali délka čáry překračuje jednu banku na displeji pokud ne jsou vybrány 2 masky (jedna pro začátek čáry, druhá pro konec) a vhodně užitou nonekvivalencí pak vytvoří krátkou čáru, která je následně vložena na pozici *LcdCache* [*index*]. Pokud délka čáry přesahuje banku, je vybrána počáteční maska, jež je vložena na pozici *LcdCache* [*index*]. Další bajty včetně koncové masky jsou vkládány iteračně na pozice *LcdCache* [*index+=102*]. Po vložení čáry jsou samozřejmě modifikovány *HiWaterMark* i *LoWaterMark* pokud je to třeba.



Obrázek 7.5 Použití funkce *LcdVLine* () pro rychlé kreslení vertikálních čar

LcdHLine (*int x1h*, *int x2h*, *int yh*, *int modeh*)

Funkce slouží pro vykreslení horizontální čáry. Na jejím počátku je ze souřadnice *y* spočtena pozice čáry v bance, a banka ve které se bude tato čára nacházet. Podle pozice v bance je z předdefinovaných masek vybrána jedna, která je následně vkládána do *LcdCache* [*index*] kde *index* odpovídá souřadnici *x1* v příslušné bance a tento postup je iteračně opakován dokud není index roven souřadnici *x2*. Po vykreslení je provedena kontrola a případná modifikace *HiWaterMark* i *LoWaterMark*.



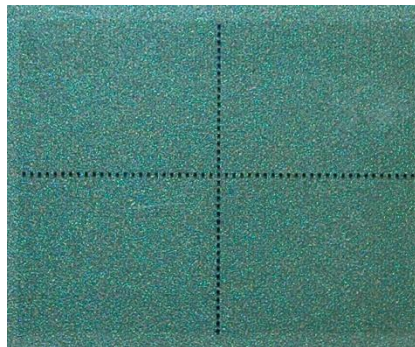
Obrázek 7.6 Ukázka funkce *LcdHLine* () pro rychlé kreslení horizontálních čar

LcdHLineDot (int x1h, int x2h, int yh, int modeh)

Funkce pro kreslení horizontální tečkované čáry. Princip je obdobný jako pro kreslení jednoduché horizontální čáry, avšak jsou zde jiná pravidla pro kreslení čáry, kdy je třeba vynechat vždy jeden bajt v *LcdCache* [].

LcdVLineDot (int xv, int y1v, int y2v, int modev)

Funkce pro kreslení vertikální tečkované čáry. Princip je stejný jako při kreslení jednoduché vertikální čáry, jsou zde však nadefinovány masky, které střídají jednotlivé bity a jejich složením vznikne tečkovaná čára.



Obrázek 7.7 Ukázka funkcí pro kreslení rychlých tečkovaných čar

LcdVRDArrow (int x, int y, int mode)

Funkce pro nakreslení poloviční pravé šipky směřující dolů, pro kreslení využívá vertikálních čar, x a y jsou souřadnice vrcholu šipky.

LcdVLDArrow (int x, int y, int mode)

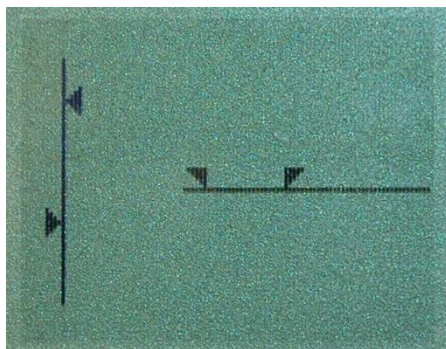
Funkce je stejná jako předchozí, ale poloviční šipka je z levé strany, šipka taktéž ukazuje směrem dolů.

LcdHRArrow (int x, int y, int mode)

Funkce pro nakreslení plné šipky ukazující směrem doprava. Vrchol šipky je na souřadnicích x a y.

LcdHLArrow (int x, int y, int mode)

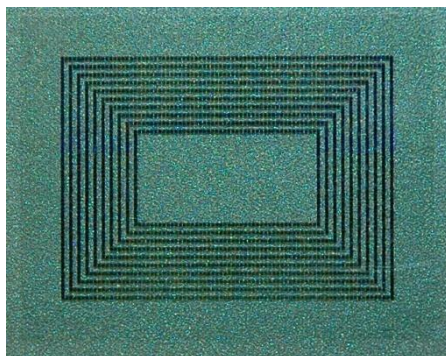
Funkce stejná jako předchozí, šipka ukazuje směrem doleva.



Obrázek 7.8 Ukázka funkcí pro kreslení šipek

LcdRectangle (int x1, int y1, int x2, int y2,int mode)

Funkce pro vykreslení obdélníku pomocí funkcí rychlých čar, kde x1 a y1 jsou souřadnice pravého horního rohu a x2 y2 jsou souřadnice levého dolního rohu. Parametr *mode* umožňuje vykreslení obdélníku, smazání obdélníku a nonekvivalenci s původním obrazem.



Obrázek 7.9 Ukázka funkce *LcdRectangle ()* pro kreslení obdélníků

LcdInit (byte addrX, byte addrY, byte first)

Tato funkce slouží k nastavení inicializačních dat při prvním spuštění displeje pokud parametr *first==1*, jinak je využívána k adresování jednotlivých částí displeje při překreslování pouze části plochy.

LcdClear ()

Funkce slouží ke smazání plochy displeje, toto je nutné například při počátečním spuštění, kdy v paměti RAM displeje není definovaná hodnota jednotlivých pixelů. Funkce vynuluje celé pole *LcdCache[]* a dále volá funkci *LcdUpdate()*, která smaže plochu displeje.

LcdCacheClear ()

Funkce obdobná předchozí, avšak bez volání *LcdUpdate()*. Používá se v případě, kdy je třeba nakreslit celou novou obrazovku přičemž je žádoucí odeslání dat na displej již s novými údaji, které se nekryjí se starými.

LcdHI ()

Funkce je využívána pro přesunutí pozice pole *LcdCache[]* do horní poloviny displeje. Tato funkce je nutná v případě mikrokontroléru, kde je potřeba snížit nároky na paměť RAM na minimum. Pole *LcdCache[]* je v tomto případě zmenšeno na polovinu. Užití není vhodné pro kreslení složitějších grafických operací.

LcdLO ()

Jedná se o obdobu předchozí funkce která umožní vykreslení spodní poloviny displeje.

LcdMI (byte bank)

Funkce umožňuje zapsání dat do libovolné banky displeje, pole *LcdCache[]* může mít v tomto případě velikost pouze jedné banky. Tato funkce je primárně určena pro mikrokontroléry s pamětí RAM pod 512B. Kdy pole *LcdCache[]* zabere pouze 100B. Tento mód je určen pouze pro výstup textu velikosti 1 s velkým omezením grafiky.

LcdXorBank (byte bank)

Funkce pro provedení rychlé nonekvivalence s daty v bance, kterou chceme změnit. Funkce je primárně určena pro zvýraznění jednotlivých položek grafického menu, či kontextové nápovědy.



Obrázek 7.10 Ukázka funkce pro rychlou nonekvivalenci jedné banky

LcdClearBank (byte bank)

Funkce slouží pro rychlé smazání jedné banky v poli *LcdCache[]*, tato funkce je nutná pro dynamické posouvání textu v bance.

LcdUpdate ()

Funkce pro inicializování přenosu informací z pole *LcdCache[]* do paměti displeje, funkce zkontroluje *HiWaterMark* i *LoWaterMark*, zdali nedošlo k posuvu mimo oblast displeje. Dále nastaví příslušné adresování pro obnovení pouze části displeje a zavolá funkci *LcdSend()*. *HiWaterMark* je vynulován, *LoWaterMark* je nastaven na počet bytů které obsahuje pole *LcdCache[]*.

LcdTimeOut ()

Funkce navázaná na časovač procesoru MSP430. Pokud do určité doby nedojde k uvolnění flagu který označuje zdali byla data již odeslána, je přenos po I2C rozhraní ukončen, proběhne reset displeje a nová inicializace a je volána funkce *LcdClear()* pro odstranění neaktuálních dat, nebo dat poškozených chybným přenosem po I2C rozhraní z plochy displeje. Funkce vrací 1 pokud došlo k chybě.

LcdSend ()

Funkce pro nastartování přenosu dat na displej, jakmile je přenos aktivní, je volána funkce *LcdTimeOut()*, která pozastaví vykonávání programu dokud není přenos ukončen. Toto pozastavení procesoru není optimální vzhledem k využití času celého systému, nicméně je brána v potaz nepříliš velká paměť RAM použitého procesoru, a tak je pole *LcdCache[]* použito přímo jako vstupní buffer pro funkci odesílání dat po I2C. Z tohoto důvodu není možné jakýmkoli způsobem měnit tato data, během odesílání, aby nedošlo k nekonzistenci.

Procesory MSP430 umí pomocí integrovaného čítače dat odeslat pouze 256 bytů, což je v případě tohoto grafického displeje, jehož plocha činí 102x80 bitů (1020 bytů), nedostačující. Proto nebyla v případě displeje použita samostatná knihovna pro odesílání dat po I2C. Obsluha přerušení je řízena softwarově. Druhý důvod pro takovouto koncepci bylo vyslání inicializačních dat přímo před vlastními daty pro zobrazení na ploše displeje, kdy tato data spolu s polem *LcdCache[]*, tvoří jeden celek.

7.1.3 Knihovna pro obsluhu SPI rozhraní (spi.c, spi.h)

Tato knihovna implementuje základní funkce pro přenos dat pomocí SPI rozhraní.

InitSpi ()

Funkce nastavuje potřebné parametry přenosu pomocí sběrnice SPI. Vzhledem k tomu, že je tato knihovna použita pouze pro komunikaci s bezdrátovým obvodem nordic, jsou některé možnosti, které SPI periferie procesorů MSP430 umožňují, nevyužity. Proto je přímo zvolena polarita a fáze hodin a SPI tak pracuje v módu 1. Frekvence hodin SCK je nastavena na 2MHz.

SpiWrite (byte data)

Funkce pro odeslání jednoho bajtu po SPI sběrnici. Dojde k přesunutí dat do vysílacího bufferu a zahájení přenosu. Zároveň je odstraněno echo kterým slave jednotka odpověděla.

SpiRead (byte addr)

Funkce pro přečtení bajtu z dané adresy, zapíše adresu po SPI rozhraní a vrací přečtený bajt.

SpiReadMulti (byte addr, byte *buffer, byte cnt)

Funkce pro přečtení datového paketu z rozhraní SPI, funkce zapíše adresu po SPI rozhraní a z této pak vyčítá jednotlivé bajty v počtu daném proměnnou *cnt*.

7.1.4 Knihovna pro UART (usart.c, usart.h)

Tato knihovna je určena pro převodník USB-nordic rfIRC35, kde je využito komunikace s obvodem FT232R.

InitUart (byte baudrate)

Inicializace přenosu se vstupním parametrem rychlost přenosu. Vybere příslušné piny na portu a nastaví přenosovou rychlost.

SendByteUart (byte data_byte)

funkce pro odeslání jednoho bajtu po sériové lince. Ihned po volání dochází k přesunu obsahu proměnné *data_byte* do vysílacího bufferu a je zahájen přenos.

SendMessage (byte *string_data)

Funkce pro odeslání řetěze znaků na sériovou linku.

Příjem dat ze sériové linky

Tato rutina je řešena obsluhou přerušení a přesunu přijatých dat do příslušného bufferu, přičemž dojde k nastavení příznaku o nově přijatých datech. Tato data mohou být obsloužena během provádění kódu.

7.1.5 Knihovna pro rozhraní reálného času (MSPClock.c, MSPClock.h)

Pro běh aplikace, ve které se předpokládá přesná znalost času nejen na úrovni obyčejných hodin, ale je nutný horizont dnů, měsíců a roků, je nezbytná implementace knihovny reálného času.

Procesory MSP430 mají možnost připojení hodinového krystalu s frekvencí 32768kHz, z kterého je možné celkem snadno odvodit sekundovou základnu hodin, přičemž přerušení těchto hodin lze svázat s několika úspornými módy procesoru. Nejvyšší úrovně úspor energie je dosaženo v módu 3, ve kterém je celý procesor uspán, a jeho probuzení je navázáno na externí přerušení, či právě zmíněné hodiny reálného času, které jsou v tomto módu aktivní.

Day (int d, int m, int y)

Funkce pro zjištění aktuálního dne ze zadaného data. Při implementaci této funkce bylo odzkoušeno několik algoritmů, avšak jejich výsledky nebyly uspokojivé, jelikož se vždy jednalo o nějaké omezení ohledně současného století, či omezení pouze na krátký horizont. Proto byl navržen vlastní heuristický algoritmus, který je založen na znalosti prvního dne nějakého přestupného roku a znalosti počtu dnů v jednotlivých měsících.

Algoritmus nejprve zjistí kolik let uplynulo od referenčního přestupného roku, dále kolik přestupných let bylo celkem v tomto období, a dále jestli se zrovna nacházíme v přestupném roce. Z těchto informací vypočítá počet dní které již uplynuly až do roku, ve kterém se aktuálně nacházíme, dále pak přičte počet dní všech měsíců které uplynuly od začátku roku a přičte aktuální den v měsíci. Pokud toto číslo vydělíme 7, pak zbytek po dělení ukazuje na den, ve kterém se právě nacházíme. 0 odpovídá pondělí, 1 úterý,6 neděle. Tato funkce vrací právě číslo dne v týdnu. Algoritmus je při použití 16

bitové aritmetiky schopen pracovat 179 let od zadaného pevného roku. V současné implementaci s pevným přestupným rokem nastaveným na rok 2004 bude pracovat tento algoritmus správně až do roku 2183.

ClockInit32k ()

inicializace kontinuálního módu časovače pro generování přerušení každou 1 s.

Obsluha přerušení od časovače reálného času

Jelikož se jedná o nízkopříkonovou aplikaci a počítá se s využitím úsporných módů procesoru, je nutné aby probuzení z úsporného módu za účelem modifikace hodinových proměnných byla co nejkratší. Proto je použita koncepce vnořování, kdy jednou za sekundu dojde k inkrementaci čítače sekund od zapnutí systému *second_counter* a zároveň pokud je proměnná *seconds < 59* dojde její inkrementaci. Pokud je v době kdy by měla být inkrementována její hodnota právě 59, je vynulována a algoritmus se propadne o úroveň níže a inkrementuje proměnnou *minutes*. V okamžiku kdy by měla být proměnná *minutes* inkrementována a její hodnota je 59, je vynulována a algoritmus se opět propadne o úroveň níže k inkrementaci proměnné *hours*. Takto je řešeno vnořování až na úroveň jednotlivých let.

Tato koncepce umožňuje zkrátit dobu potřebnou na udržování hodin reálného času při probuzení procesoru z úsporného režimu, kdy obsluha proměnné *seconds* se provádí každou sekundu, obsluha proměnné *minutes* včetně zjištění jestli inkrementovat či vynulovat pouze jednou za minutu a tak dále. Čas, po který procesor pracuje v aktivním režimu po probuzení, je tedy zcela minimální.

7.1.6 Knihovna pro obsluhu tlačítek (*buttons.c*, *buttons.h*)

Bezdrátový termostat je osazen čtyřmi tlačítky, které při stisku stáhnou k nulovému potenciálu vstupy mikrokontroléru. Tyto vstupy jsou v klidovém stavu taženy k úrovni logická „1“ pomocí čtyř Pull-Up rezistorů.

ButtonInit (byte slope)

Funkce pro inicializaci tlačítek, včetně výběru hrany pro generování přerušení podle zadaného parametru.

Obsluha přerušení od tlačítek

Pokud dojde k přerušení od brány 1, ke které jsou tlačítka připojena, je podle vektoru přerušení nastaven příslušný flag pro tlačítko které bylo stisknuto (například při stisku tlačítka 3 dojde k nastavení flagu *buton3_pressed = 1*). Zároveň je vždy nastaven flag *any_button = 1*, který značí, že došlo ke stisku nějakého tlačítka. Vynulování těchto flagů je ponecháno na obsluhující aplikaci.

ButtonWaitRelease (byte button)

Funkce která čeká dokud nedojde k uvolnění tlačítka, jehož číslo bylo zadáno pomocí parametru *button*. Pokud je parametrem *button* předána hodnota 5, funkce čeká na uvolnění všech tlačítek.

Button1Repeat (unsigned int repeat_interval, unsigned int start_repeat)

Tato funkce je stejná pro každé tlačítko proto bude probrána pouze u jednoho. Parametr *start_repeat* je doba po jejíž uplynutí dojde k prvnímu opakování. Parametr *repeat_interval* je čas opakování stisků při držení stisknutého tlačítka. Při zavolání tato funkce zkontroluje zda byl nastaven příslušný flag *button1_pressed==1*, pokud toto neplatí funkce vrátí 0. Pokud však podmínka platí dojde k vynulování tohoto flagu, funkce vrátí 1 a inkrementuje proměnnou kterou srovnává se zadaným parametrem *start_repeat*. Pokud je tlačítko stále stišťeno, funkce vrátí 0 dokud vnitřní čítač není roven hodnotě *start_repeat*. V tomto okamžiku opět vrátí hodnotu 1 a začíná inkrementovat proměnnou pro opakování. Jakmile v tato proměnná dosáhne hodnoty *repeat_interval*, funkce opět vrátí 1.

Tato funkce je implementována i pro stisk libovolného tlačítka a opakovací sekvence běží až do okamžiku uvolnění všech tlačítek.

7.2 Popis nadstavbových knihoven

7.2.1 Knihovna pro obsluhu bezdrátového modulu Nordic nrf905

Tato knihovna slouží pro nastavování parametrů bezdrátového obvodu Nordic nrf905 a pro základní bezdrátovou komunikaci mezi jednotlivými bezdrátovými zařízeními navrženými právě na s tímto obvodem. Tato knihovna je postavena jako nadstavba knihovny *spi.c*, pomocí které probíhá konfigurace a přenos uživatelských dat mezi procesorem a obvodem nrf905.

CsnDelay (int j)

Jedná se o pomocnou funkci, která je užita pro synchronizaci signálu CS (chip select) a zahájení respektive ukončení přenosu dat po SPI rozhraní.

NrfInit ()

Funkce pro inicializaci bezdrátového obvodu, nataví příslušné brány pro řídicí signály a pošle pomocí sběrnice SPI inicializační data. Obvod nrf905 umí odeslat najednou 1 až 32 bytů, ovšem díky implementaci části linkové vrstvy musí být délka odesílaných dat známá, proto je při inicializaci obvodu nastavena pevná délka dat, definovaná jako `SIZE_OF_DATA_TO_SEND`.

NrfProg ()

Funkce pro nastavení programovacího módu obvodu nrf905.

NrfSleep ()

Funkce pro usnutí obvodu, v tomto režimu nepracuje ani oscilátor, je však možné zapisovat inicializační data do registrů obvodu.

NrfSet (byte ownaddr, byte channel, byte power)

Funkce pro nastavení parametrů přenosu. Nastavení vlastní fyzické adresy je dané parametrem *ownaddr*. Obvody nrf905 umí až 4B adresy, v tomto případě je za účelem

urychlení přenosu a tím snížení spotřeby využito pouze adresy dálky 1B. Data od ostatních jednotek budou přijímána na kanálu daném parametrem *channel*, přičemž je možné nastavit jeden z 256 kanálů v daném pásmu. Nastavení vysílacího výkonu je možné ve 4 krocích 10dBm, 6dBm, -2dBm, -10dBm.

NrfSend (byte addr, byte data [SIZE_OF_DATA_TO_SEND])

Funkce odešle vložená data o délce *SIZE_OF_DATA_TO_SEND* jednotce s adresou danou parametrem *addr*. Aby mohla být data přijata je potřeba aby jednotka které přijímá i jednotka která vysílá měly při inicializaci nastaven stejný kanál. Funkce také kontroluje, zda se podařilo data odeslat.

NrfRecv ()

Tato funkce čeká do okamžiku, kdy obvod nrf905 nastaví signál DR, který říká, že byla přijata data se správnou hlavičkou a CRC kódem a je možno je vyčíst z přijímacího registru. Tuto funkci je možné volat přímo v přerušení vyvolaném příchozím signálem DR. V okamžiku, kdy je signál DR aktivní, jsou data přesunuta pomocí rozhraní SPI do paměti mikrokontroléru.

7.2.2 Knihovna pro funkce grafického menu (menu_lib.c, menu_lib.h)

Tato knihovna implementuje nadstavbové funkce rozšiřující možnosti knihovny LCD.c .

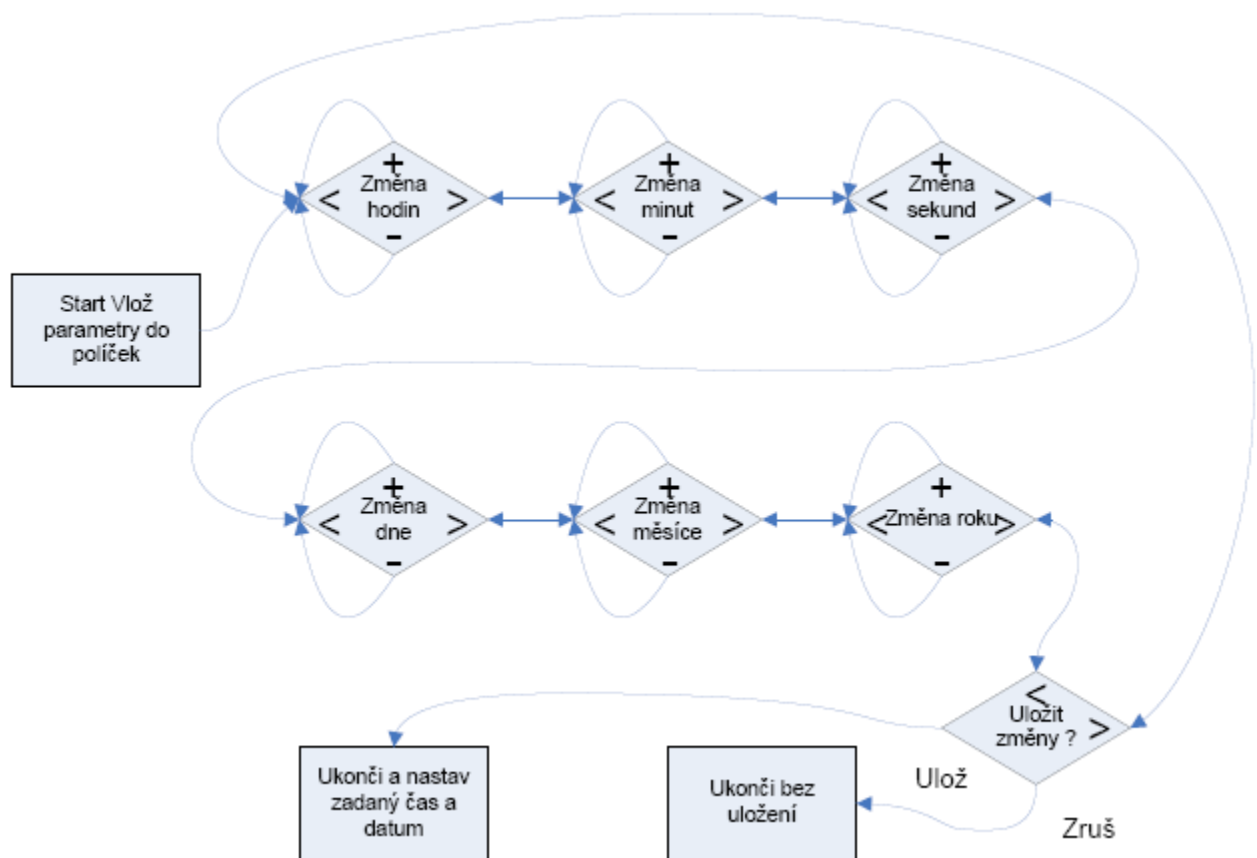
SetTimeDate(int set_seconds, int set_minutes, int set_hours, int set_days, int set_months, int set_years)

Jedná se o grafickou komponentu, která umožňuje zadávání data a času. Vstupními parametry lze přednastavit aktuální čas či datum, nebo přednastavit na nějakou defaultní hodnotu.

Komponenta pracuje v několika smyčkách, mezi kterými se přeskakuje pomocí příznaků pro přepnutí do jiné smyčky. V průběhu každé smyčky je dotazován stav tlačítek pomocí funkcí *Button1Repeat ().....Button4Repeat ()*, a na základě jejich aktivity jsou nastavovány příslušné události. Základní obrazovku tvoří nápověda, která je zobrazena v horní části displeje. Pod nápovědou je tabulka jejíž první řádek tvoří

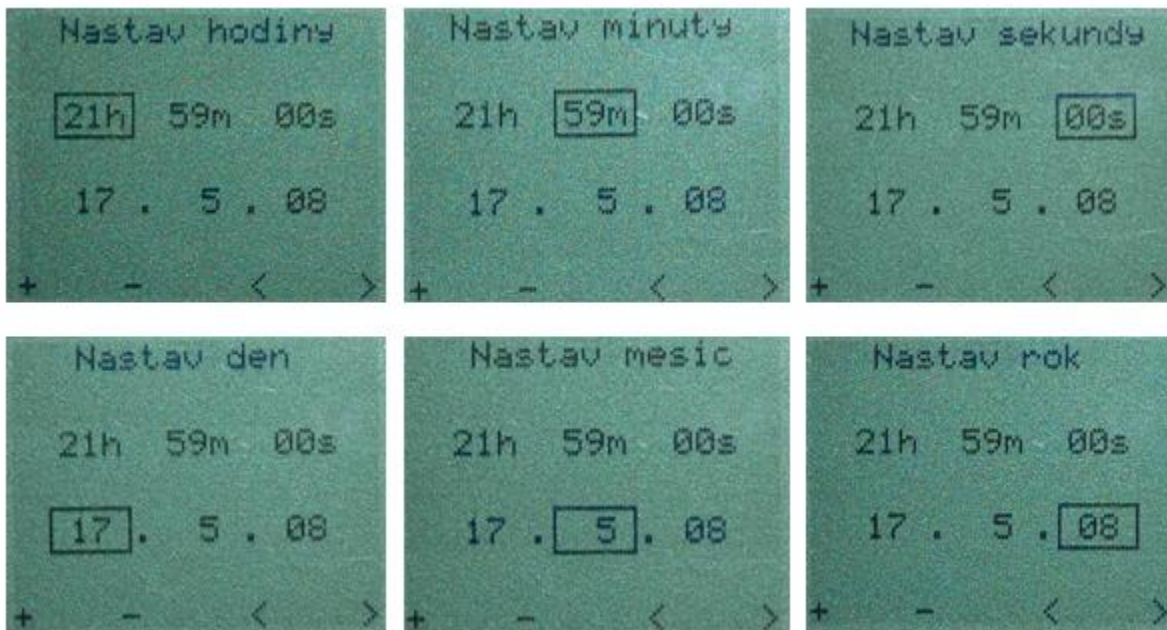
čas a druhý řádek datum. Základní obrazovku zakončuje kontextová nápověda, která zobrazuje funkci jednotlivých tlačítek.

Po spuštění komponenty je první aktivní smyčka pro nastavení hodiny, což signalizuje obdélníček kolem údaje „hodina“, jehož hodnota je nastavena pomocí parametru *set_hours*. Změna hodnoty daného políčka je možná pomocí tlačítek s kontextem + či -. Přejechod do další smyčky, resp. přechod na další údaj realizují tlačítka s kontextem < či >. Všechny stavy a přechody mezi nimi ukazuje Obrázek 7.11. Skutečné ukázky zobrazení na displeji ukazuje Obrázek 7.12 a Obrázek 7.13. Pokud je například aktuální stav ve smyčce pro nastavení minut (kolem údaje s minutami je obdélníček), stisk tlačítka s kontextem < přeskočí do smyčky pro editaci hodin (obdélníček je nyní kolem údaje s hodinami). Pokud je stisknuto tlačítko s kontextem > přeskočí přejde program do smyčky pro editaci sekund.

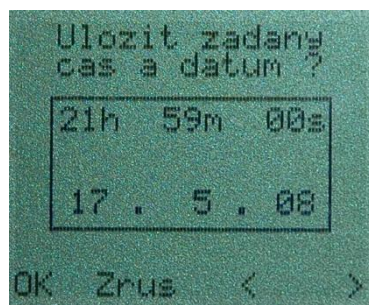


Obrázek 7.11 Schéma komponenty pro nastavení data a času

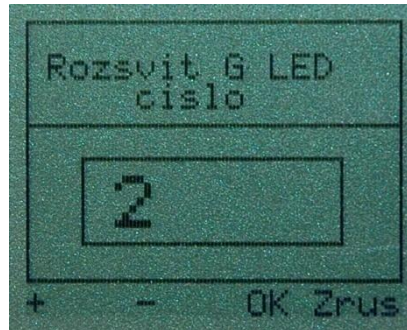
Při volbě uložit jsou aktualizovány všechny proměnné času a data, a zároveň je volána funkce $Day(int\ d, int\ m, int\ y)$, které jsou jako parametry vloženy zadaný den, měsíc a rok. Tato funkce vrátí aktuální den v týdnu, podle kterého je ještě aktualizována proměnná day_of_week .



Obrázek 7.12 Jednotlivá okna pro editaci času a data



Obrázek 7.13 Nabídka pro uložení data a času



Obrázek 7.15 Komponenta pro zadávání číselné hodnoty

ViewTime(int x, int y, int font_size)

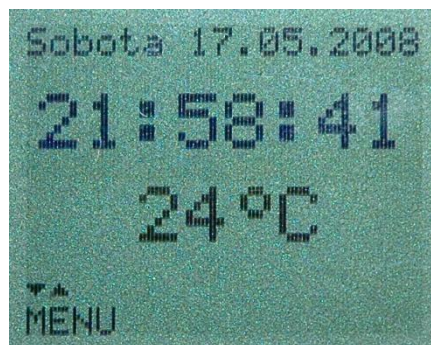
Funkce pro zobrazení času ve formátu HH:MM:SS na pozici danou parametry x a y . Velikost fontu daná parametrem $font_size$ může být volena 1 nebo 2. Funkce zobrazuje čas podle proměnných, které jsou aktualizovány pomocí knihovny *MSPClock.c*.

ViewDayOfWeek(int x, int y)

Funkce která převede informaci z proměnné day_of_week na jméno aktuálního dne a vypíše jej na pozici danou parametry x a y .

ViewDate (int x, int y)

Funkce pro výpis aktuálního data na pozici danou parametry x a y ve formátu DD.MM.RRRR



Obrázek 7.16 Náhled pro komponenty zobrazení času, dne v týdnu a data

7.2.3 Knihovna pro nastavení topné křivky (`tempcurve.c`, `tempcurve.h`)

Jelikož se jedná o aplikaci pro individuální řízení teploty je nutné, aby bylo možno zadat topný režim. S přihlédnutím k možnostem samotného termostatu existují 2 možnosti jak křivku zadat.

- Zadání pomocí tabulky – tato možnost je velice jednoduchá, avšak nepraktická a nepřiliš přehledná pro větší počet změn v topné křivce.
- Zadání pomocí grafického prostředí – toto řešení nabízí možnost vytvoření intuitivní aplikace pro zadání či editaci topné křivky. Pro aplikaci v termostatu rflRC25 je zvolena právě tato varianta.

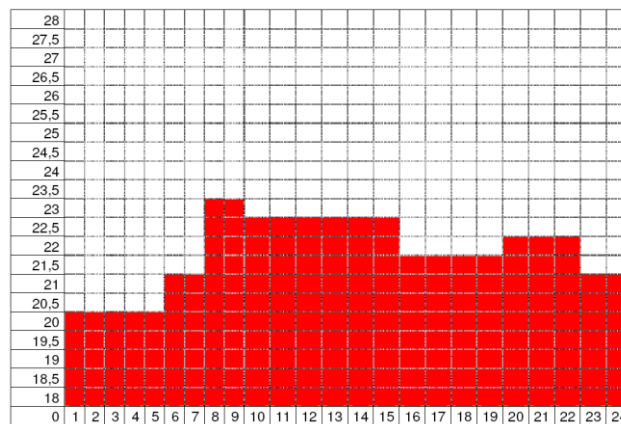
Aby bylo možné realizovat topné programy, které budou složeny ze sedmi topných křivek (pro každý den jedna), je nutné zamyslet se nad složitostí topných programů a nad celkovými nároky jak na samotné jednotky systému, tak nároky na datové objemy přenášené v bezdrátové síti. Z tohoto důvodu je nutné stanovit možnosti nastavení topných křivek a definovat jasná omezení:

- Počet změn teploty v jednom dni maximálně 8
- Možnost nastavení změny po 15 minutách (vychází 24 hodin x 4 změny = 96, což se vejde do 1 byte). Vzhledem k setrvačnosti budov by stačilo i 30 minut, avšak přesnější nastavení může být užitečné pro přímotopné zdroje tepla.
- Nastavení v rozsahu standardně 5÷30 °C s přesností 0,5 °C (Měření a samotná regulace však musí být minimálně s přesností 0,1 °C, je dokázané že člověk vnímá i změny ±0,5 °C a nepřesná regulace narušuje tepelnou pohodu). Tento údaj obsadí druhý byte. Topný program pro 7 dní tedy celkově obsadí 14 bytů

*SetTempCurve (byte New, byte *temperature_curve)*

Jedná se o grafickou komponentu, které umožňuje intuitivní nastavení denní topné křivky. Pro nastavení celého týdne bude tato komponenta v podmenu s názvem každého dne. Tato koncepce dále umožní například i nastavování stejného programu pro celý týden, nebo zvláštní pro týden a víkend atd. Možný příklad graficky znázorněné topné

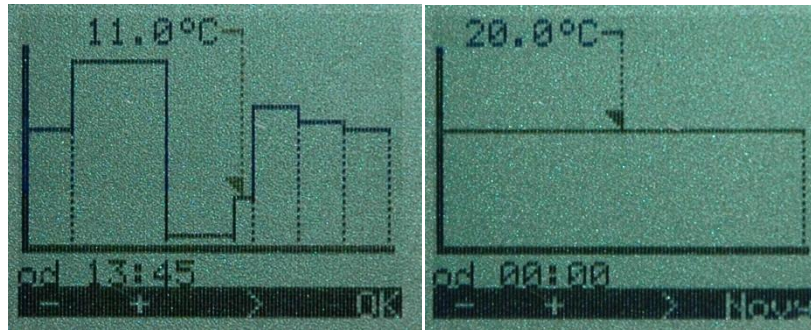
křivky ukazuje Obrázek 7.17. Na časové ose je čas od 0 do 24 hodin, vlevo je teplotní stupnice.



Obrázek 7.17 Příklad graficky zadané topné křivky

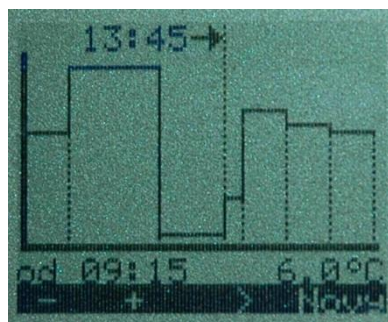
Komponenta respektuje omezení která byla kladena na topné programy. Parametr *new* udává, zda se jedná o vytváření nové křivky, či editaci již dříve vytvořené. Celý průběh tvoří hlavní smyčka, ve které pracují iteračně tři podsmyčky. Zjednodušený vývojový diagram pro nastavení topné křivky ukazuje Obrázek 7.21.

1. Smyčka na nastavení teploty. Dojde k načtení času předchozí změny a času následující změny, přičemž při vstupu do této smyčky je obnovena celá grafická obrazovka, včetně kompletní topné křivky pro celý den. V úseku předchozí změny a následující změny je nyní možno pomocí tlačítek s kontextem **+** a **-** měnit teplotu od 5 do 30 °C. Pro intuitivnější ovládání je nad křivkou vynesena údaj o nastavované teplotě spolu s šipkou, která bliká a ukazuje směrem k aktuálnímu úseku křivky. Stiskem tlačítka s kontextem **>** přejde program do druhé smyčky (nastavení času konce úseku viz. Obrázek 7.19). Stiskem tlačítka s kontextem **OK**, přejde program do třetí smyčky (potvrzovací okno viz. Obrázek 7.20). Náhled první smyčky ukazuje Obrázek 7.18



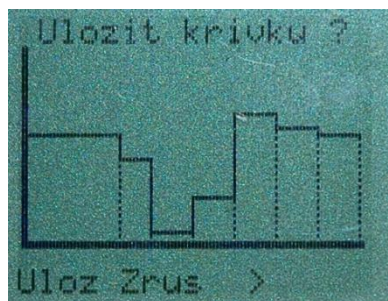
Obrázek 7.18 Ukázka obrazovky pro zadávání teploty

2. Smyčka pro nastavení konce úseku. V tomto okamžiku je již nastavena teplota, a je nutné nastavit konec daného úseku. Aktuální hodnota konce je načtena z předložených dat. Na displej je zobrazen čas aktuální změny (konce úseku) a od údaje o čase k této změně bliká šipka. Pomocí tlačítek s kontextem + a - je nyní možné měnit ubírat či přidávat čas. Pokud při snižování času dojde k překrytí času konce změny s časem začátku změny, úsek je automaticky odstraněn, a program přechází na nastavení teploty předchozího úseku. Pokud naopak při zvyšování času dojde k překrytí konce změny s koncem následujícího úseku, je následující úsek automaticky vypuštěn. V tomto režimu je u tlačítka 4 zobrazen kontext *Nový*, což umožňuje přidat nový úsek, který jehož začátek je od aktuální pozice do poloviny následujícího úseku. Teplota nového úseku je defaultně nastavena na 20°C. Tento úsek lze přidat pouze v případě, že délka následujícího úseku je alespoň 1 hodina a není již vyčerpáno 8 změn v celé křivce. Stisk tlačítka s kontextem > přeskočí do první smyčky pro editaci teploty následujícího úseku.

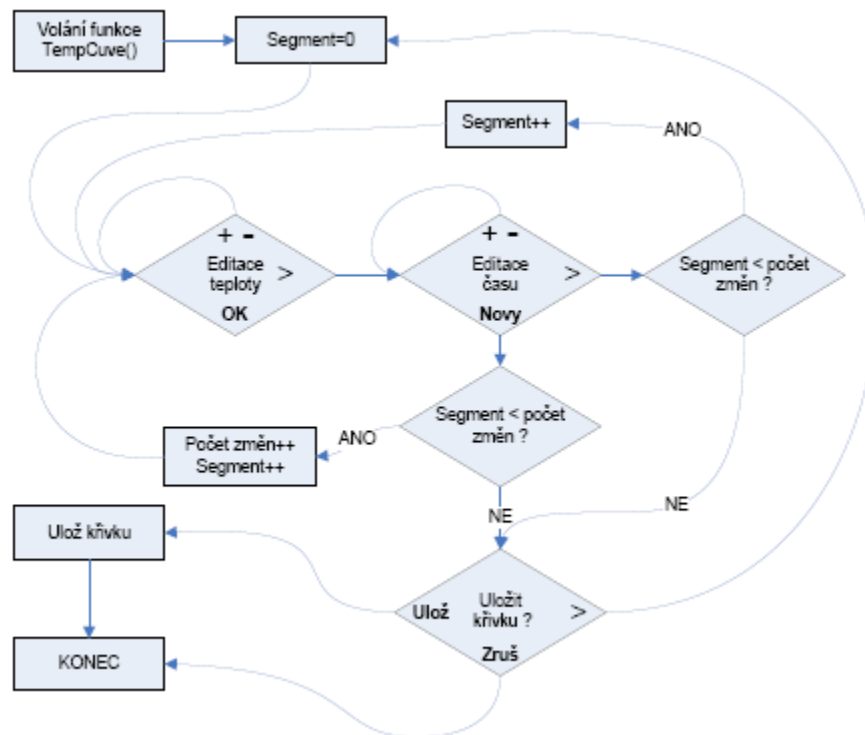


Obrázek 7.19 Ukázka obrazovky pro zadávání času konce úseku

3. Smyčka potvrzovacího okna. Pokud je během zadávání topné křivky stisknuto tlačítko s kontextem *OK*, nebo je stisknuto tlačítko s kontextem *>* při změně teploty úseku, který má koncový čas právě 24 hodin. Je program přesměrován do této smyčky, která slouží k potvrzení uložení stiskem tlačítka s kontextem *Ulož*, nebo k opuštění zadávání topné křivky bez uložení stisknutím tlačítka s kontextem *Zruš* nebo k úpravě zadané křivky stiskem tlačítka s kontextem *>*. Při stisku tohoto tlačítka se program vrátí na úpravu teploty prvního úseku.



Obrázek 7.20 Ukázka potvrzovacího okna



Obrázek 7.21 Zjednodušené schéma komponenty pro nastavení topné křivky

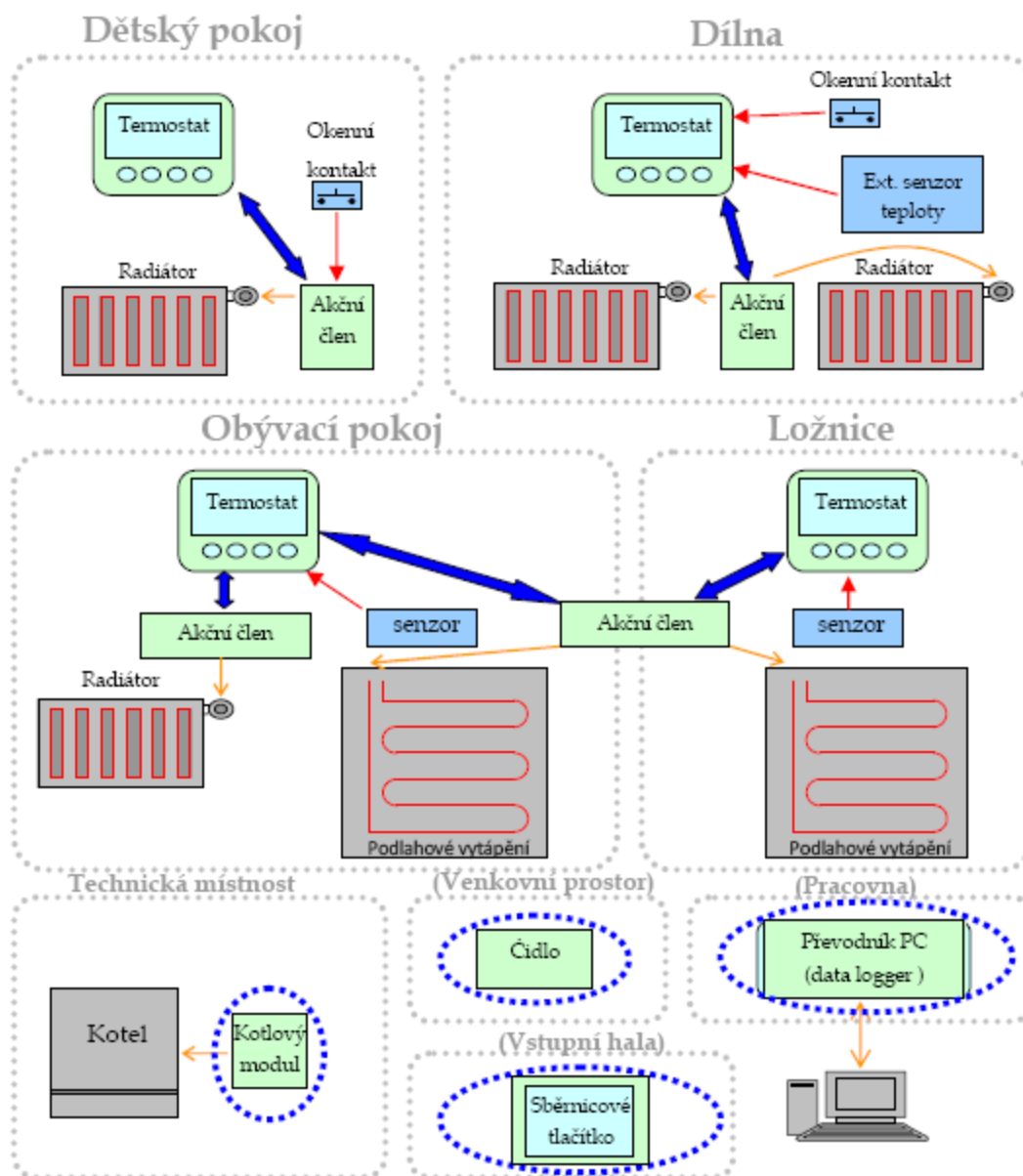
Kapitola 8

Teoretický návrh bezdrátového protokolu

8.1 Struktura bezdrátové sítě

Bezdrátová síť bude řízena mastery třídy 1 a 2 podobně jako u sítě Profibus. Jednotky budou nezávisle pracovat v malých celcích „Termostat-akční člen“, případně „Termostat-více akčních členů“ (tento celek tvoří jednu Slave jednotku) . Nad nimi bude dohlížet Master třídy 1 (převodník USB-Nordic), který může sám šířit broadcastové zprávy o stavu domu (např. odchod poslední osoby atd.) a zároveň bude monitorovat síť (kde docházejí baterie atd.)

Pro sestavování sítě a sledování, nastavování parametrů či topných křivek bude sloužit aplikace v prostředí .NET (Master třídy 2 resp. konfigurační aplikace viz. Kapitola 4.6), která bude s distribuovanými jednotkami propojena pomocí Mastera třídy 1. Tato aplikace může běžet na běžném PC, či malém průmyslovém počítači, přičemž činnost systému bude na této aplikaci nezávislá. V budoucnu pak tato aplikace může umožnit i propojení systému vytápění s jinými systémy například pomocí sítě ethernet. Možnou konfiguraci systému pro běžnou domácnost ukazuje Obrázek 8.1.



Obrázek 8.1 příklad možné konfigurace systému bezdrátové regulace teploty

- ↔ Tyto šipky označují bezdrátovou komunikaci point to point (např. Termostat-akční člen)
- Označení prvků, které jsou globálně zapojeny do systému (např. tlačítko pro generování události - poslední uživatel opustil dům), prvky mohou vysílat bezdrátový broadcast pro ostatní jednotky

→ → Reprezentují fyzické propojení. Například silové vodiče pro spínání přímotopných těles

8.2 Řešení kolizí při komunikaci

Pokud se nějaké zařízení pokusí navázat spojení, odposlechne nejprve nosnou. V případě, že bude odposlech negativní, může začít vysílat. V opačném případě vygeneruje náhodný čas, po kterém opět odposlechne nosnou. Pokud bude opět komunikovat někdo jiný bude se tento postup opakovat až do doby, kdy bude nosná volná. Vzhledem k velice krátkým zprávám, vysílaným řádově po jednotkách minut, je pravděpodobnost opakovaného odposlechu nosné v důsledku obsazenosti kanálu velice nízká.

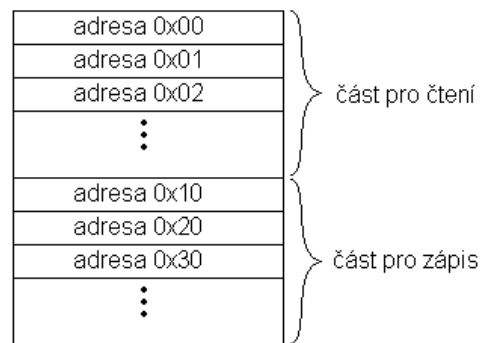
Přesto se může stát, že si dvě (či více) jednotek ve stejnou chvíli ověří, že nikdo nekomunikuje a začnou komunikovat. Následkem této situace nebude nikdo schopen přijmout smysluplnou zprávu. Tento stav jednotky, které se snažily komunikovat poznají podle toho, že do určité doby neobdrží potvrzení o přijetí správných dat. Jednotky proto vygenerují náhodný čas, po kterém se pokusí navázat spojení znovu podle principu, který se používá v případě obsazeného kanálu.

8.3 Uspořádání dat v jednotlivých jednotkách

Aby byl protokol dostatečně univerzální, nejsou jednotlivá data v jednotkách prezentována v závislosti na jejich významu. To znamená, že protokol nepracuje s dotazy na konkrétní hodnotu něčeho, co je jednotka schopna poskytnout, ale pouze s dotazy na obsah určitého místa v definované paměti. Pokud např. převodník rfIRC35 potřebuje, aby mu termostat rfIRC25 poslal stav svých baterií. Požádá ho o obsah paměti, o které ví, že potřebný údaj obsahuje. Stačí tedy, aby převodník rfIRC35 věděl o jaký typ jednotky jde a z toho bude vyplývat, co je možné od ní požadovat a kde v její paměti se to nachází.

Ke konfiguraci jednotlivých jednotek v systému bude sloužit soubor *.XML, který bude výstupem konfiguračního softwaru viz. Kapitola 4.6. Tato aplikace je vyvíjena mimo tuto práci a nemá smysl vyvíjet implementaci tohoto protokolu aniž by byl znám přesný výstup z této konfigurační aplikace.

Obecná struktura paměti dat každé jednotky bude obsahovat dvě základní části: část pro čtení a část pro zápis. Příklad možného uspořádání paměti ukazuje Obrázek 8.2.

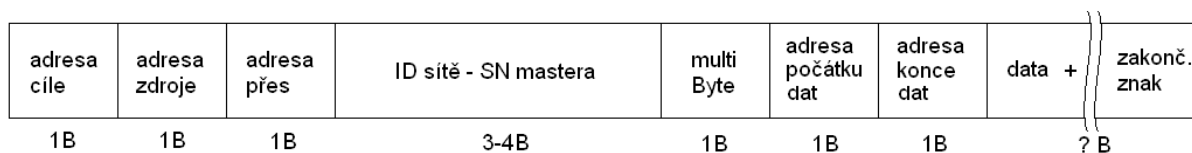


Obrázek 8.2 Příklad možnosti využití části paměti pro informace o jednotce

V části pro čtení budou uložena data, která je jednotka schopna získat pomocí svých prostředků (např. teplota, vlhkost, informaci o otevřených oknech atd.) Obsah této části paměti bude možné předat ostatním stanicím v síti. Druhá část, určená pro zápis, bude sloužit např. k nastavování a konfiguraci zařízení. Pokud bude do této paměti něco zapsáno, bude to použito pro řízení funkce stanice. (V závislosti na obsahu této paměti mohou být např. nastavovány pracovní režimy, spínány relé, nastavovány prahy měřených veličin atd.)

8.4 Formát přenášených dat

Jednotlivé zprávy přenášené po síti budou mít strukturu, kterou ukazuje Obrázek 8.3, toto je první struktura. Je možné že bude ještě pozměněna. (např. přidání ještě jedné či více adres pro retranslaci)



Obrázek 8.3 Struktura dat přenášených v síti

Význam jednotlivých částí datového paketu

adresa cíle – identifikátor stanice v síti, pro kterou jsou data určena

adresa zdroje – identifikátor stanice v síti, která zprávu odeslala

adresa přes – identifikátor stanice v síti, přes kterou se má zpráva odesílat, nutné pro retranslaci

ID sítě – identifikátor sítě shodný s výrobním číslem převodníku USB-Nordic slouží k jednoznačnému určení, komu zpráva náleží. ID sítě je nutné pro případ dvou a více sousedních sítí v dosahu. Protože adresy jednotek jsou jedinečné pouze v rámci jedné sítě, mohlo by mezi více sítěmi docházet k doručování zpráv jednotkám, kterým nejsou určeny.

multi byte – tento Byte má více významů

3 bity slouží pro identifikaci významu zprávy (požadavek na data, odpověď na požadavek, potvrzení o přijetí, start autodetekce, odpověď na autodetekci, detekce ok...)

5 bitů slouží pro přenášení více paketů, které k sobě patří (1 příznak, že ještě něco přijde, zbytek číslování paketů)

adresa počátku dat – adresa v paměti dat, kde začínají požadovaná data nebo kde začíná místo pro zápis.

adresa konce dat – adresa v paměti dat, kde končí požadovaná data nebo kde začíná místo pro zápis.

data – samotná data včetně ukončovacího znaku

8.5 Adresace jednotek v síti

Adresování jednotlivých stanic v síti probíhá pomocí 1-Bytové adresy, kterou „master“ přidělí všem jednotkám „slave“ při jejich prvním připojení k bezdrátové síti. Aby bylo toto přidělování možné, musí mít všechny jednotky „slave“ implicitně nastavenou stejnou adresu, pomocí které si s „masterem“ domluví tu jedinečnou. Jednotky „master“ musí mít všechny trvale stejnou adresu.

Pro sestavení sítě a přidělení adres je nutné aby všechny zařízení byly v „inicializačním režimu“ Do tohoto režimu se dostanou vždy po připojení napájecího napětí nebo pomocí vlastní funkce zařízení.

Princip systému autodetekce

Funkce je založena na principu hledání unikátního ID za pomoci půlení intervalu. Pro autodetekci jsou potřeba následující speciální pakety:

DETECTION_START - Vymaž si příznak, že tě master už zná (posílá se jako broadcast)

RESPONSE_WHILE (PATTERN,B) - Odpověz svým ID, pokud se shoduje s šablonou *PATTERN* od bitu *B* výše

DETECT_OK (ID) - Nastav si příznak, že tě master už zná

Na obr. 4 je vývojový diagram, který znázorňuje princip algoritmu autodetekce včetně systémem ošetření náhodných chyb. Součástí vyobrazení jsou následující proměnné:

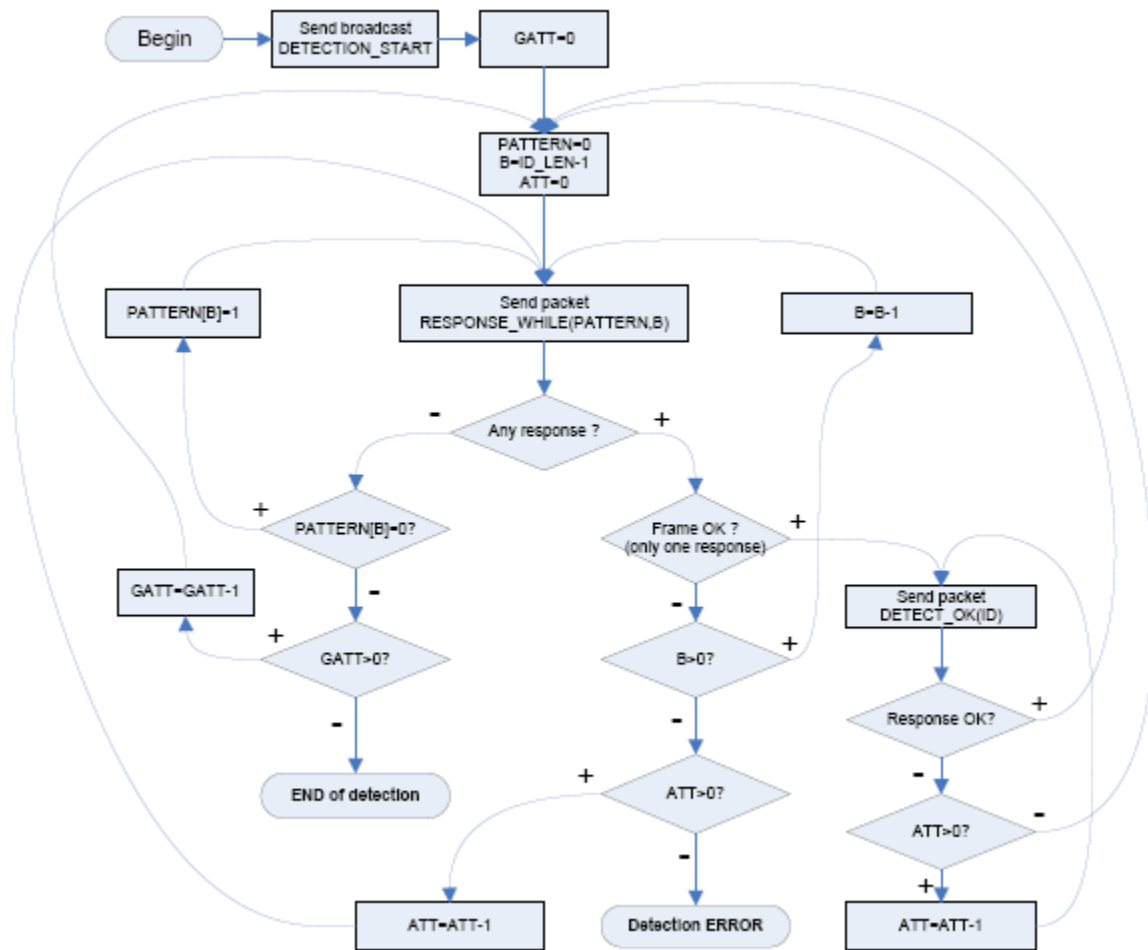
ID_LEN - počet bitů ID (předpokládaných 24 bitů)

GATT - (**G**lobal **att**empts) celkový počet pokusů při neúspěchu

ATT - (**A**ttempts) počet pokusů posílání paketu *RESPONSE_OK*, případně při naražení na úplnou šablonu, na kterou pořád chodí chybné odpovědi (nastane např. když existují 2 jednotky se stejným ID)

Princip autodetekce ukazuje pomocí půlení intervalů ukazuje Obrázek 8.4. Tato metoda je však relativně složitá, a je otázkou, zda při skutečné implementaci nebude použito jen

ohlašování v náhodných časech, které by pro předpokládaný maximální rozsah sítě kolem 100 jednotek mohlo fungovat také spolehlivě.



Obrázek 8.4 Princip autodetekce pomocí půlení intervalu

Kapitola 9

Reálná měření na systému

9.1 Rozbor a měření spotřeby

Jelikož bezdrátový termostat je zařízení napájené z baterií, nabízí se otázka, jak dlouho vlastně vydrží s bateriemi pracovat? Pro komerční nasazení by bylo třeba aby jednotka vydržela pracovat v systému minimálně 1 rok. Když se vezme v potaz, že kvalitní alkalické články velikosti AAA mají kapacitu kolem 1500mAh. Je tedy nutné rozpočítat tuto hodnotu na horizont jednoho či dvou let a tím dostaneme maximální přípustnou spotřebu, kterou může jednotka odebírat. Tabulka 9.1 uvádí maximální průměrný proud, který může bezdrátový termostat odebírat aby mohl pracovat deklarovanou dobu.

Kapacita AAA (mAh)	Maximální spotřeba pro výdrž 1 rok (mA)	Maximální spotřeba pro výdrž 2 roky (mA)
1500	0,171	0,085

Tabulka 9.1 Maximální přípustné spotřeby bezdrátového termostatu

Reálný výpočet spotřeby ukazuje Tabulka 9.2. Jsou zde brány v úvahu jednotlivé stavy zařízení a četnost jejich výskytu v průběhu dne. Například obnovení displeje je potřeba

provést každých 60s pro udržení aktuálního času (aktualizace minut) na displeji. V tomto režimu je zařízení v aktivním stavu, avšak není podsvícen displej. Naměřená spotřeba zařízení v tomto režimu je 1,4mA a obnovení trvá kolem 10ms. Z těchto údajů lze snadno spočítat, že na obnovování displeje se spotřebuje každý den 0,005 mAh. U ostatních údajů jsou úvahy podobné.

Tabulka 9.2 ukazuje, že podle předpokladu je možné, aby navržený termostat mohl pracovat relativně dlouhou dobu. Tato doba se může změnit podle toho jak často uživatel bude provádět změny topných programů, nebo v případě častějšího probouzení samotných jednotek za účelem odesílání aktuálnějších informací o teplotě.

	Spotřeba při aktivitě (mA)	doba aktivity za jeden den (s)	četnost události (s)	doba jedné události (s)	Spotřeba za 1 den (mAh)
Spotřeba při vysílání	30	1,843	300	0,00032	0,0153
spotřeba na příjmu	10	14,4	300	0,02	0,04
Spotřeba aktivního režimu s podsvícením	8	17,143	604800	120	0,038
Spotřeba aktivního režimu bez podsvícení	1,4	14,4	60	0,01	0,005
Spotřeba LPM3 displej aktivní polovina	0,08	86352,214	-----	-----	1,918
Celková spotřeba za 1 den (mAh)					2,0179
Celková spotřeba za 1 rok (mAh)					736
Maximální předpokládaná výdrž při AAA 1500mAh (roky)					2,036

Tabulka 9.2 Výpočet spotřeby a reálné výdrže baterií

9.2 Rozbor vlastností antén a dosahu

Jelikož se jedná o bezdrátové zařízení, dosah je jeden z parametrů, který může zásadním způsobem ovlivnit funkci celého systému. V tomto případě se jedná o použitou frekvenci 868 MHz, která má vlnovou délku 345,6 mm. V úvahu byly brány 2 typy antén:

- Dipól s kondenzátory po zmenšení rozměrů . Tato anténa byla převzata z materiálů od firmy Nordic přičemž vycházela velice malých rozměrů (12x12mm). Tato anténa měla díky kondenzátorům v dipólu sice správné impedanční přizpůsobení, ovšem díky rozměrům velice malou efektivní plochu. A neukázala se jako vhodná pro aplikaci v systému regulace teploty.
- Prutová čtvrtvlnná anténa s desimetrizačním členem. Byly použity 2 antény tohoto typu, jedna sestávala z drátu o délce 86,4mm (čtvrtina délky vlny) a druhá byla komerčně prodávaná. Tato komerční anténa byla menší díky vinutí drátu do spirály. Toto řešení má výhodu v menších rozměrech, nevýhoda je že její charakteristika je frekvenčně závislá a tak zejména na okrajích použitého pásma má mírně menší zisk.

9.2.1 Praktické orientační měření

Z praktických měření se neosvědčil smyčkový dipól leptaný přímo na plošném spoji, se kterým byl dosah ve volném prostranství kolem 60m a v obytných prostorech 1 zeď tlustá kolem 0,5m.

Mnohem lépe dopadly v tomto ohledu čtvrtvlnné prutové antény, jejich dosah byl i na okrajích pásma téměř srovnatelný. Tyto antény umožnily dosah 150m ve volném prostranství a v budovách umožnily průchod signály přes dvě 0,5m silné zdi a ještě cca 20m za nimi.

Tato měření jsou pouze orientační avšak pro praktické užití mají svoji vypovídací hodnotu, proto jim byla věnována tato část práce.

Kapitola 10

Výroba krytu pro termostat rFIRC25

V této práci se vyvíjen produkt, u něhož se předpokládá komerční využití, z tohoto důvodu se bylo potřeba zamyslet i nad úpravou krabičky, která je běžně k dostání, avšak nelze ji přímo použít pro plnohodnotné vestavění navrženého termostatu rFIRC25.

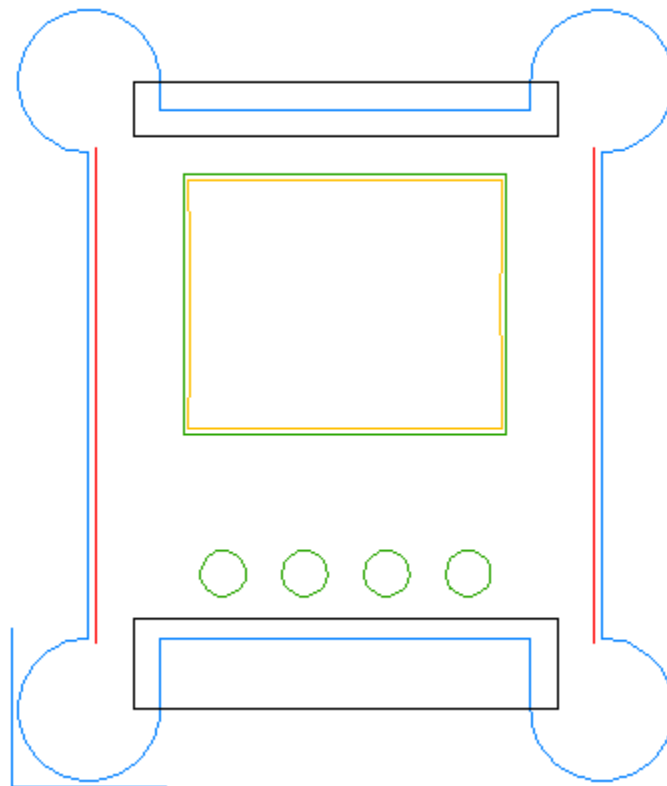
10.1 Výkresy a popis frézování krytu ABB Time

V dnešní době jsou již poměrně hojně rozšířeny NC obráběcí stroje, které umožňují rychle a velice přesně obrábět výrobky podle dat, které je možno získat přímo z elektronických výkresů dané součásti. Těmto datům se říká G-kódy a obsahují mj. informaci o trajektorii nástroje pomocí něhož je daná součást obráběna. Pro přípravu G-kódu z vektorových výkresů byl použit software ArtCam Insignia.

10.1.1 Frézování přípravku pro upevnění krytu

Pro frézování byl vybrán kryt, který firma ABB nabízí pro montáž komerčních čidel. Tento kryt má oproti obyčejné záslepce výhodu v tom, že již má na dolní a horní hraně průduchy pro odvětrání vnitřního prostoru. Použití tohoto krytu usnadní a tím pádem i zlevní práce nutné k úpravě krytu. Obrázek 10.1 ukazuje výkres, který byl použit pro frézování přípravku pro uchycení krytu i pro frézování vlastního krytu.

Celý přípravek je tvořen MDF deskou. Nejprve byla vyfrézována vanová prohlubeň znázorněná modrou čarou. Tato prohlubeň byla vyfrézována asi do hloubky 25mm. Tato prohlubeň tvoří opěrné stěny pro vlastní kryt, který je tímto vystředěn. Dále byly zafrézovány opěrné plošky do hloubky 10mm, tyto jsou znázorněny černými obdélníky. O tyto plošky bude následně opřen kryt a pod ním zbývá místo nad odpadávající špony po frézování. Aby bylo postup možné kdykoli zopakovat, je nutné správné zkalibrování os X a Y pro daný přípravek. K tomuto účelu slouží malý zápich ve tvaru písmene L v levém dolním rohu. Na výkrese označen modrou barvou. K přípravku byly ještě dodatečně přišroubovány 2 upínky pro pevné uchycení krytu.

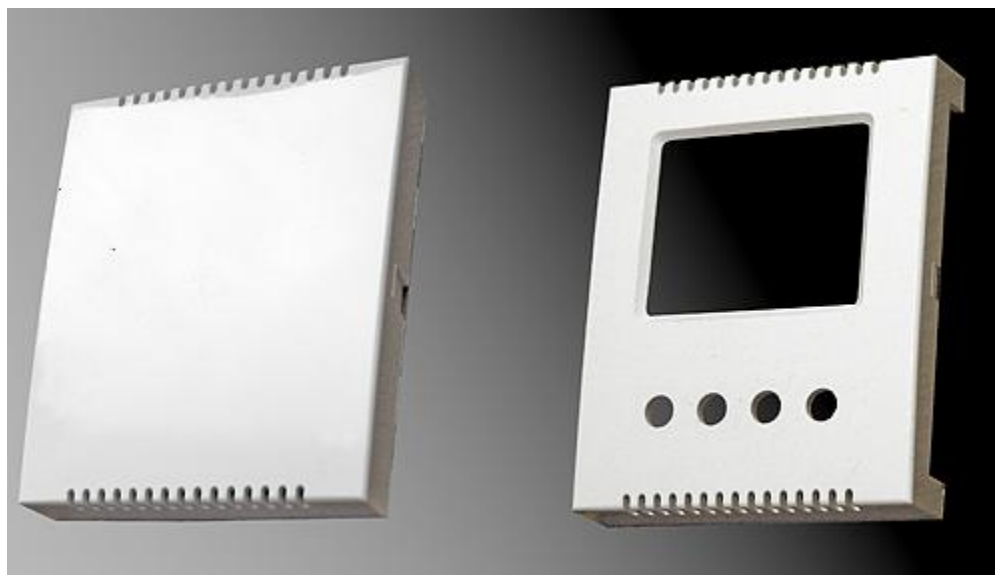


Obrázek 10.1 Výkres pro přípravu G-kódů

10.1.2 Frézování vlastního krytu

Jelikož přípravek i frézování vlastního krytu vychází z jednoho výkresu, je možno použít kalibrační značky (L v pravém dolním rohu) pro nulové souřadnice X a Y. Nejprve jsou

do krytu z horní strany vyfrézovány otvory pro displej a tlačítka, které jsou označeny zelenou barvou jak ukazuje Obrázek 10.1. Dále je vyměněna fréza s průměrem 2mm za speciální frézu s vrcholovým úhlem 90° (hrot má tvar písmene V), a touto frézou je podle trajektorie označené okrovou barvou vyrobena fazetka kolem otvoru pro displej. Tato trajektorie kopíruje 3D povrch čidla tak, aby byla fazetka rovnoměrná v celém obvodu otvoru pro displej. Poslední operace nutná k dokončení krytu je vyfrézování místa ze spodní strany, které je nutné pro umístění baterií. Na výkresu jsou tyto trajektorie označeny červenou barvou. Použitá hloubka je 5mm od vrchní plochy přípravku (nulová hodnota osy Z). Tvar krytu před a po úpravě ukazuje Obrázek 10.2.



Obrázek 10.2 Výsledný kryt před a po úpravě

Kapitola 11

Zhodnocení

Tato diplomová práce navazovala na předcházející bakalářskou práci, ve které byla řešena „Řídící jednotka pro regulaci teploty“, v té době se jednalo především o základní návrh hardware. Tato jednotka posloužila v této práci jako zařízení pro implementaci základního softwaru, a zároveň jako základ pro vývoj bezdrátového termostatu rfIRC25.

Jeden z cílů práce byl návrh a implementace bezdrátového protokolu. Ačkoliv byl protokol navržen, jeho implementace byla odložena, kvůli neexistujícímu softwaru na konfigurování celé bezdrátové sítě IRC systému. Ve vzniklém prostoru byl pak kladen větší důraz na jednotlivé HW komponenty a na vlastní software pro jednotlivé jednotky.

Byly navrženy celkem 3 nové jednotky a jedna byla převzata z projektu CEPOT s názvem „Sběr dat ze senzorů fyzikálních veličin s možností bezdrátové komunikace“. Při oživování jednotlivých modulů byl největší problém s bezdrátovým obvodem, jehož pouzdro není vhodné pro ruční osazování. V současné době tento problém přetrvává na jednotkách rfIRC25. Pro případný další vývoj by tyto bezdrátové moduly byly nejspíše nahrazeny modernějšími od firmy HOPE MICROELECTRONICS, které implementují celé zapojení včetně desimetrizačního členu na malém DPS. Takto zapouzdřený obvod je již možno snadno osadit. Všechny nově navržené jednotky splňují požadavky na zástavbu systému bezdrátové regulace teploty do běžných domácností. Záležitostmi kolem modifikace jednotky termostatu pro použití sběrnice RS485 se tato práce

nezabývala, jelikož tato jednotka je vyvíjena odděleně s podporou protokolu uLAN. Nicméně prostorové uspořádání bude odvozeno od jednotky bezdrátového termostatu rfIRC25.

Ze softwarové části byl kladen důraz na oživení potřebných periférií daného mikrokontroléru MSP430, dále byla implementována podpora externích čidel OneWire, a ostatních náležitostí potřebných pro práci těchto jednotek. Velký důraz byl kladen na vývoj softwaru, který umožní intuitivní a rychlé nastavení topné křivky. Při vývoji byla důkladně promyšlena koncepce ovládání pomocí 4 tlačítek, jimiž je jednotka limitována kvůli velikosti. Další nezbytnou součástí bylo implementace knihovny reálného času včetně zjištění dne v týdnu. V budoucnu by nad těmito knihovnami měla být postavena komponenta, která bude realizovat příslušný topný program, včetně PI regulace pro řízení jednotlivých akčních členů. S příchodem protokolu OpenTherm se nabízí možnost implementace řízení kotlových jednotek. V tomto řízení by mohla být zahrnuta informace z jednotlivých pokojů celého objektu a např. pomocí váhových funkcí nastaven požadovaný optimální výkon kotle.

Tato práce je pouze začátkem pro vývoj celého kvalitního systému, pro individuální regulaci teploty. V okamžiku kdy bude hotov konfigurační software, může být implementován bezdrátový protokol a je otázka zda nevyužít možností dnes již existujícího protokolu pro bezdrátové sítě kterým je například ZigBee, či systému pro tvorbu bezdrátových aplikací TinyOS.

Poděkování

Děkuji Ing. Pavlu Němečkovi za vedení mé diplomové práce, za čas strávený při konzultacích a za věcné připomínky a rady.

Děkuji Ing. Pavlu Burgetovi za podporu a formování myšlenek v programu CEPOT.

Poděkování patří také všem kteří mě podporovali ve studiu a zvláště pak rodině.

Použité materiály

- [1] Texas Instruments, *Mixed Signal Microcontroller (Rev. D)*, [online], 31 Mar 2005, <http://www-s.ti.com/sc/ds/msp430f167.pdf>
- [2] Texas Instruments, *Programming a Flash-based MSP430 Using the JTAG Interface*, [online], 30 Sep 2005, <http://focus.ti.com/lit/an/slaa149a/slaa149a.pdf>
- [3] Texas Instruments, *MSP430x1xx Family User's Guide (Rev. F)*, [online]. , 28 Feb 2006 <http://focus.ti.com/lit/ug/slau049f/slau049f.pdf>
- [4] Nordic Semiconductor, *nRF905_rev1_4*, [online], 2006 http://www.nordicsemi.no/files/Product/data_sheet/nRF905_rev1_4.pdf
- [5] EM Microelectronics, *EM6125 datasheet*, [online], 2005, http://www.emmicroelectronic.com/webfiles/Product/LCDdriver/EM6125_DS.pdf
- [6] ST Microelectronics, *L6920 datasheet*, [online], 2005, <http://www.st.com/stonline/products/literature/ds/7320.pdf>
- [7] Dallas Semiconductor, *DS18B20 datasheet*, [online], 2006, <http://pdfserv.maxim-ic.com/en/ds/DS18B20.pdf>
- [8] Underwood, S.: *mspgcc manual*, [online], 2003, <http://mspgcc.sourceforge.net/manual/>

Použitý software

Orcad	10.0,	http://www.cadence.com
Mspgcc	3.2.3,	http://mspgcc.sourceforge.org
Eclipse	3.3.0,	http://www.eclipse.org
Doxygen	1.5.5,	http://www.doxygen.org
WinARM	4.1.0,	http://winarm.scienceprog.com
AutoCAD	2008,	http://www.autodesk.com
ArtCam Insignia	3.6,	http://www.artcaminsignia.com

Přílohy

Označení	Pouzdro + popis	hodnota
C1	SM/C_0805	3,3nF
C10,C11,C27	SM/C_0805	33pF
C12,C15,C17-C21,C29	SM/C_0805	100nF
C13	SM/C_0805	2.2uF
C2	SM/C_0805	10nF
C22	SM/C_0805	10nF
C24	SM/C_0805	3,9pF
C25	SM/C_0805	3,9pF
C26	SM/C_0805	2,2pF
C3,C4	SM/C_0805	22pF
C30	SM/DO214AA_21	100uF
C5,C6	SM/C_0805	12pF
C9	SM/C_0805	4,7nF
D1-D6	SIP/TM/L.200/2	LED 2mA
D7,D8	SIP/TM/4MM/2	SB120
D9	SIP/TM/4MM/2	1N5391
J6	MINI USB SMD	CON5
J7	TELE/TM.310FACE/6X6	RJ11
J8	SIP/TM/2MM/7	HEADER 7
J9	NAP KON R2,1	RCA JACK
L1	SM/C_0805	12nH
L2,L3	SM/C_0805	18nH
R8,R9,R10,R11,R12,R13	SM/C_0805	560R
R15	SM/C_0805	4K7
R16	SM/C_0805	22K
R6	SM/C_0805	10M
R7	SM/C_0805	200K
R14	SM/C_0805	100K
S2	SIP/TM/L.300/3	SWITCH 1X2
U2	NRF905 NEW	nRF905
U4	MSP F167 NEW	MSP430 F16x
U5	SOG.65M/28/WG8.20/L10.16	FT232RL
U6	SOIC 108	AT45D 1Mbit
U7	TSSOT 16	MAX3221
U8	SOG.050/8/WG.244/L.200	LE33CD
U9	SIP/TM/L.3,2MM/3	DS18B20
Y3	KRYSTAL_5X3	16Mhz
Y4	KRYSTAL_5X3	4Mhz

Příloha A.1 Rozpiska součástek pro převodník rfIRC35

Komponenta	Pouzdro + popis	Hodnota
BT1	2 x Bateriový klip keystone AAAA	NO.55
BT2	2 x Bateriový klip keystone AAAA	NO.55
C1	0603	22pF
C2	0603	22nF
C3	0603	33pF
C4	0603	33pF/3,3nF
C5	0603	33pF
C6	0603	4,7nF
C7	0603	10nF
C8	0603	33pF
C9	0603	3,9pF
C10	0603	3,9pF
C11	0603	3,9pF
C12	0603	1uF
C13	0603	33pF
C14	0603	33pF
C15	0603	100nF
C16	0603	100nF
C17	0603	100nF
C18	0603	100nF
C19	0603	100nF
C20	SMD Tantal velikost C	47uF
C21	0603	100nF
C22	0603	100nF
C23	SMD Tantal velikost C	47uF
C24	0603	100nF
C26	0805	2.2uF
C27	0603	100nF
D5	SOT23 2x dioda	BAT54S
J1	Pinová lišta SMD 2mm x 10pinů	
J2	Pinová lišta SMD 2mm x 10pinů	
J3	Dutinková lišta SMD 2mm x 10pinů	
J4	Dutinková lišta SMD 2mm x 10pinů	
J5	MOLEX 1mm x 10 pinů	
J10	ARK1550V/1	
J6	ARK1550V/1	
J7	ARK1550V/1	
J8	ARK1550V/1	
L1	SMD DL22-10	10uH
L2	0805	12 nH
L3	0805	12 nH
L4	0805	12 nH
Q1	SOT23 TSM2301CX	FET P
Q2	SOT23 TSM2301CX	FET P
R1	0805	22K
R2	0805	22K

Příloha A.2 Rozpiska součástek pro bezdrátový termostat rfiRC25

Komponenta	Pouzdro + popis	Hodnota
R3	0805	22K
R4	0805	22K
R5	0603	22K
R6	0603	10M
R7	0603	100K
R13	0603	4K7
R14	0603	100K
R15	0603	4k7
R16	0603	4k7
R17	0603	4K7
R19	0603	4K7
R19	0603	10M
R20	0603	4M7
R21	1206	100R
R22	0603	470R
R23	0603	10K
R24	0603	10K
R25	NTC 10K	
S1	SMD spínač TACTMS-24N	
S2	SMD spínač TACTMS-24N	
S3	SMD spínač TACTMS-24N	
S4	SMD spínač TACTMS-24N	
U10	SO8 lineární stabilizátor	LE33CD
U2	QFN32 bezdrátový obvod	Nordic nRF905
U4	PM (PQFP 64) procesor	TI MSP430 F1610
U6	TSSOP8 měnič napětí	ST L6920D
U8	EM6125 podsvícený displej	
U9	Sensirion čidlo vlhkosti	SHT11
Y1	Krystal 32.768KCFPX-56R	32 768 Hz
Y2	Krystal 5x3mm	16Mhz

Příloha A.2 Rozpiska součástek pro bezdrátový termostat rfiRC25

Komponenta	Pouzdro + popis	Hodnota
C1	kondenzator foliový 15x18x5mm 400V	100nF
C2	Kondenzator keramický 400V	5nF
C3	kondenzator foliový 15x18x5mm 400V	100nF
C4	SMD 0805	100nF
C5	kondenzator elektrolytický radialni 5mm	220uF
C6	SMD 0805	2uF
C7	SMD 0805	2uF
C8	SMD 0805	M1
C9	SMD 0805	2M
C10	kondenzator tantalový pouzdro A	10M
C11	kondenzator elektrolytický radialni 5mm	220uF
C13	SMD 0805	100nF
C14	SMD 0603	M1
C15	SMD 0603	M1
C16	SMD 0805	2M
C17	kondenzator tantalový pouzdro A	10M
C18	SMD 0603	4.7 pF
C19	SMD 0805	10 nH
C20	SMD 0603	33pF
C21	SMD 0603	22p
C22	SMD 0603	22p
C23	SMD 0603	5.6pF
C24	SMD 0603	3,9pF
C25	SMD 0603	5.6pF
C26	SMD 0603	10nF
C27	SMD 0603	33pF/3,3nF
C28	SMD 0603	33pF
C29	SMD 0603	4,7nF
C30	SMD 0603	22pF
C31	SMD 0603	22pF
C32	SMD 0603	33pF
D1	Dioda usměrňovací 1000V 1A SOD80	
D2	Dioda usměrňovací 1000V 1A SOD80	
D3	Dioda zenerova 5,5V SOD80	5,5V
D4	Dioda usměrňovací 1000V 1A SOD80	
D5	Dioda zenerova 5,5V SOD80	5,5V
D6	Dioda usměrňovací 1000V 1A SOD80	
D7	Dioda usměrňovací 1000V 1A SOD80	
J3	Dutinková lišta 2,54x2	
J4	Dutinková lišta 2,54x2	
J5	Pinová lišta 2,54x2	
J6	Pinová lišta 2,54x2	
J7	ditinková lišta 2x7	
L1	Tlumivka raiální 7mm	10mH
L2	SMD 0805	22 pF
L3	SMD 0805	12 nH

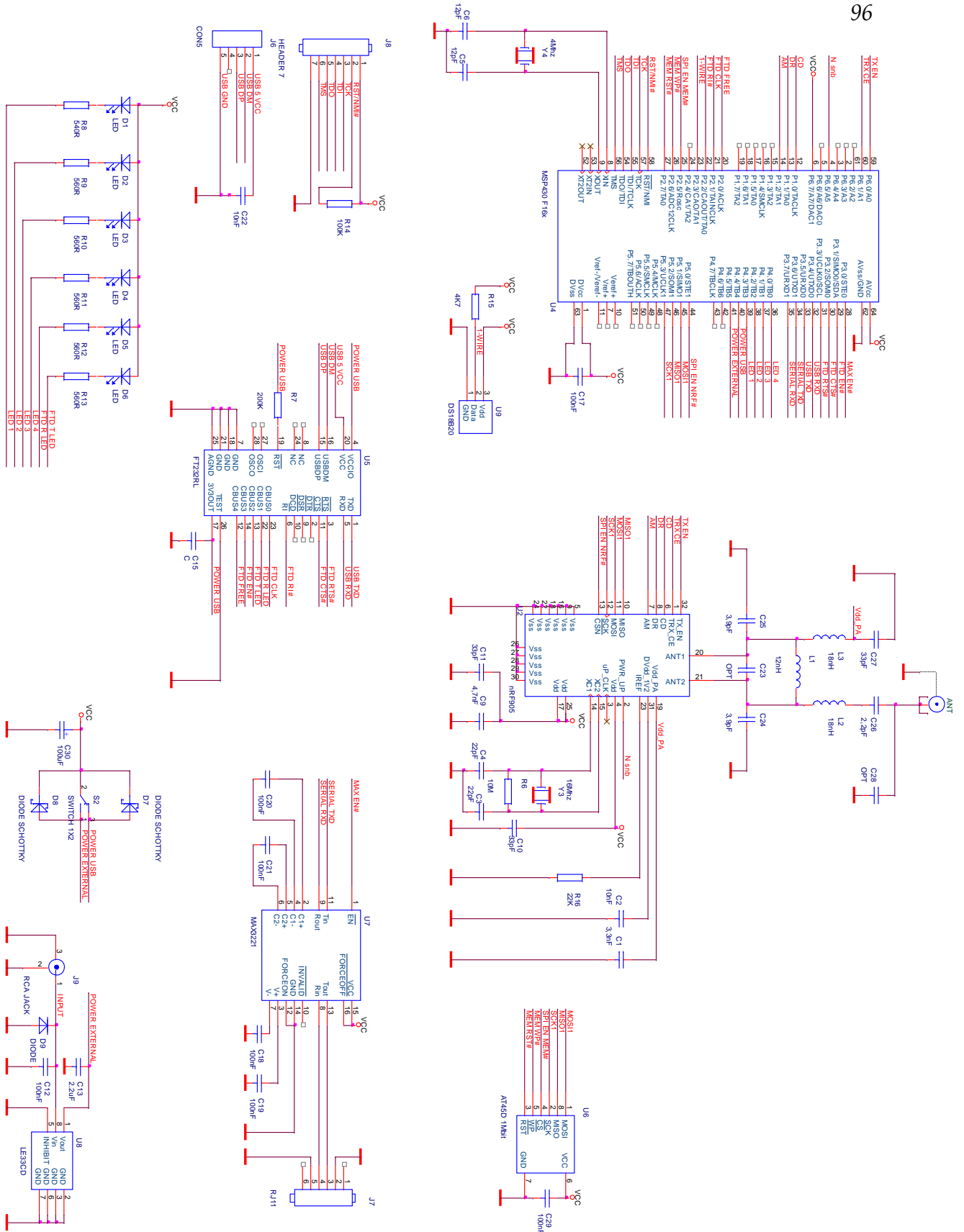
Příloha A.3 Rozpiska součástek pro bezdrátový akční člen rfIRC10

Komponenta	Pouzdro + popis	Hodnota
L4	SMD 0805	12 nH
Q1	Transistor NPN SOT23	BC846BL
Q2	N kanál Mosfet TO220	IRF840
Q3	Transistor NPN SOT23	BC546
R1	SMD 0805	330R
R2	SMD 0805	10R
R3	SMD 0805	100K
R4	SMD 0805	1K
R5	SMD 1206	470K
R6	SMD 0805	1K
R7	Rezistor 1W SIP	2,2R
R8	SMD 0603	10k
R9	SMD 0603	22K
R10	SMD 0603	10M
R11	SMD 1206	470K
R12	SMD 0603	100K
RE1	Relé 5V 16A/250VAC	
T1	Transistor NPN digitální SOT23	BCR 108
U1	CNY 17-3 optočlen DIP6	CNY-17
U2	Stabilizátor napětí 1,8V SOT23	1V8 sot23
U3	Stabilizátor napětí 3,3V SOT23	3V3 sot 23
U4	Mikroprocesor LPC2103	LPC2103FBD48
U5	Supervizor napětí 3,15V SOT23	MCP130
U6	bezdrátový trnsceiver nrf905 QFN32	nRF905
V1	Usměrňovací můstek 400V MINIDIL	S250
Y1	SMD KRYSTAL_5X3	14,7456Mhz
Y2	SMD KRYSTAL_5X3	16Mhz

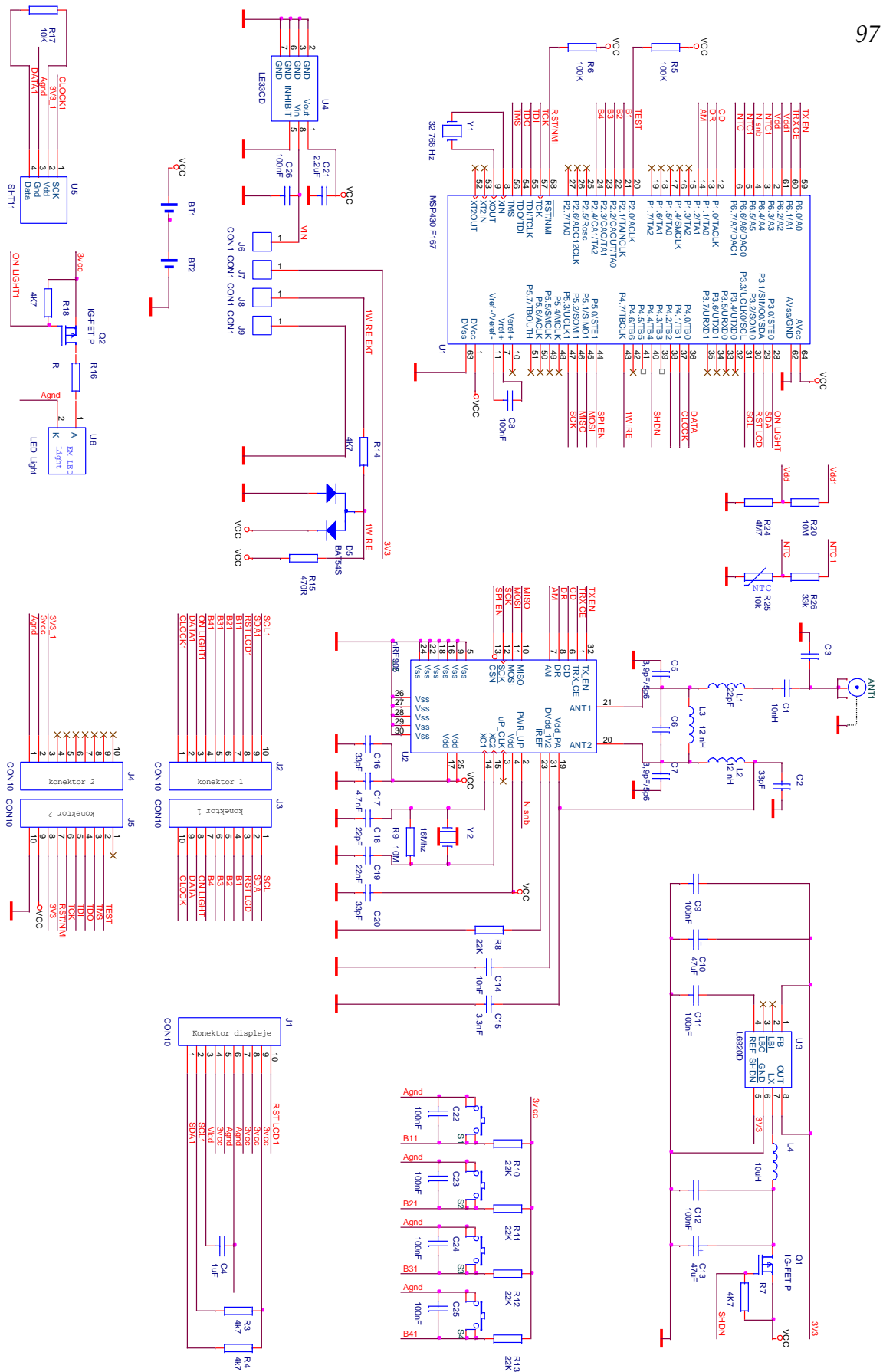
Příloha A.3 Rozpiska součástek pro bezdrátový akční člen rfIRC10

Komponenta	Pouzdro + popis	Hodnota
C1	SMD 0805	10nF
C2	SMD 0805	100nF
C4	Tantalový kondenzátor velikost A	100uF
C5	SMD 0805	100nF
C6	SMD 0805	2.2uF
C7	SMD 0805	100nF
C8	SMD 0805	10uF
D1	SMD 0805	
D2	SMD 0805	
D3	SMD 0805	
D4	SMD 0805	
D5	Schottkyho dioda SOD80	
J1	USB konektor typ B do PCB	
J2	konektor vidlice 2,54x10 dvouřadý 90°	
Q2	Tranzistor NPN SOT23	BC846BL
Q5	Mofet P kanál SOT23	
R1	SMD 0805	200K
R2	SMD 0805	47K
R3	SMD 0805	560R
R4	SMD 0805	560R
R5	SMD 0805	560R
R6	SMD 0805	560R
R9	SMD 0805	100K
R10	SMD 0805	4K7
S3	prepinac 2,54mm	
T1	Tranzistor NPN digitální SOT23	BCR 108
U1	Převodník USB/UART	FT232RL
U2	Stabilizátor napětí 3,3V SO8	LE33CD
Program	Tlacidlo 6x6mm SMD	DTSM-6
Pushbutton	Tlacidlo 6x6mm SMD	DTSM-6
Reset	Tlacidlo 6x6mm SMD	DTSM-6

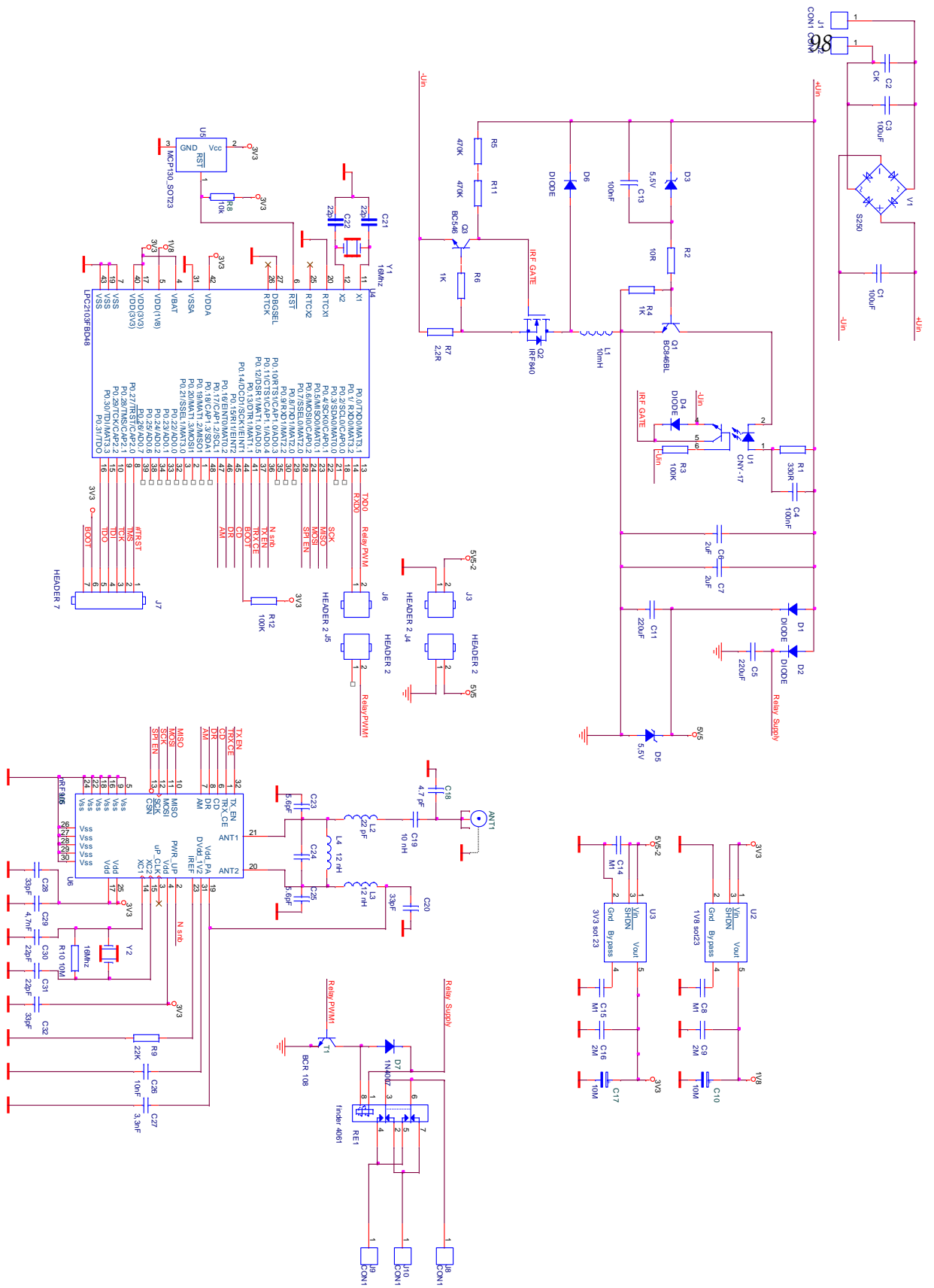
Příloha A.4 Rozpiska součástek pro USB programátor procesorů LPC21xx



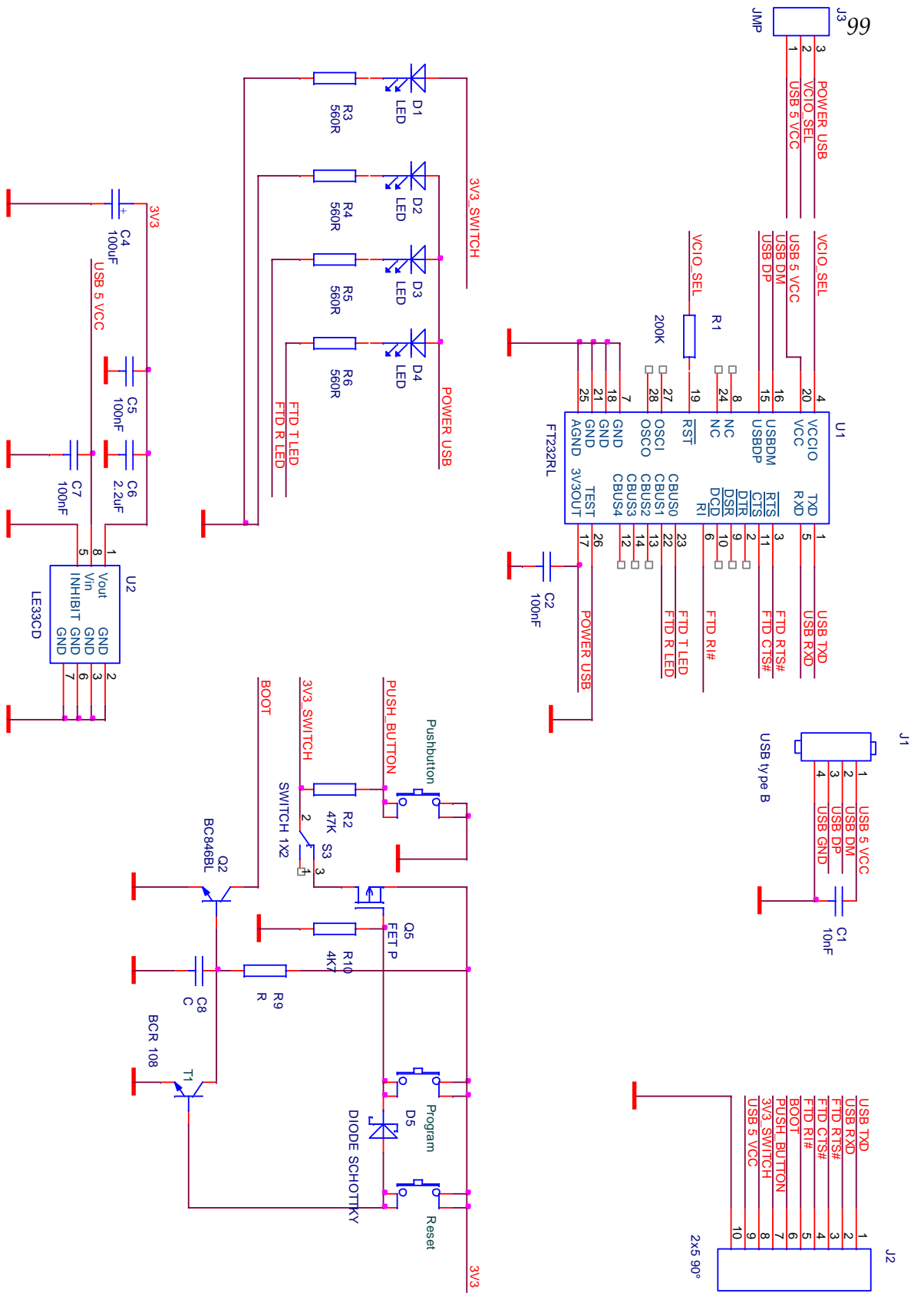
Príloha B.1 Schéma převodníku USB-Nordic rfIRC35



Príloha B.2 Schéma bezdrátového termostatu rfIRC25



Príloha B.3 Schéma bezdrátového akčného členu rFIRC10



Príloha B.4 Schéma programátoru pro procesory LPC21xx