České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídící techniky

Diplomová práce

Řídicí systém pro ovládání klimatizace a vytápění železničních vozů

Vypracoval: Bc. Michal Sluštík
Vedoucí práce: Ing. Tomáš Levora

2013

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne ……. …….

Podpis

Anotace

Nabízený komfort při cestování osobním vlakem má velký vliv na spokojenost cestujících. Jedním z prvků, které tento komfort zajišťují, je klimatizace v železničním voze. Má za úkol zajistit tepelnou pohodu ve všech kupé, po celou dobu jízdy vlaku.

Tématem této diplomové práce je návrh a realizace systému klimatizace pro železniční vozy řady ABmz. Práce na základě analýzy požadavků navrhuje strukturu řídicího systému a architekturu hardwaru, včetně softwarového vybavení mikrokontrolérů. Velkou pozornost práce věnuje zejména návrhu a implementaci komunikace mezi jednotlivými celky systému.

Důraz je kladen především na odolnost a spolehlivost celého řešení, protože se jedná o vývoj reálného projektu klimatizace, který nastoupí do každodenního provozu.

Annotation

The comfort offered whilst travelling on a passenger train has a large impact on passenger satisfaction. Air conditioning in a carriage is one of the elements that provide this comfort. It has to ensure a pleasant temperature in all of the carriage compartments in course of the entire journey.

The topic of this master's thesis is a design and application of an air conditioning system for the ABmz-series carriages. Based on a requirements analysis, this thesis designs the structure of the air conditioning control system and its hardware architecture, including microcontroller software equipment. Particular attention is paid to design and implementation of communication between the individual system components.

Design emphasis is put on the stability and reliability of the solution, as it is a project development of a real air conditioning system, which will be employed in daily operation.

Obsah

1 Úvod....1
2 Popis řešeného problému....2
2.1 Princip klimatizace....2
2.2 Současný stav....3
2.3 Návrh řešení....5
2.3.1 Modulární vs. monolitické řešení....9
3 Návrh hardwaru....11
3.1 Požadavky na hardware....11
3.2 Hlavní řídicí jednotka....16
3.2.1 Struktura řídicí jednotky....16
3.2.2 Procesorová deska....17
3.2.3 Návrh hardwaru....19
3.3 Jednotka v kupé....31
3.3.1 Struktura jednotky v kupé....31
4 Komunikace....33
4.1 Požadavky na komunikaci....33
4.2 Sběrnice LIN (Local Interconnect Network)....34
4.2.1 Vlastnosti sběrnice....34
4.2.2 Fyzická vrstva....35
4.2.3 Formát rámce....36
4.3 Sběrnice CAN (Controller Area Network)....37
4.4 Vlastnosti sběrnice....37
4.4.1 Fyzická vrstva....38
4.4.2 Formát rámce....39
4.5 Sběrnice RS485....40
4.5.1 Vlastnosti sběrnice....40
4.5.2 Fyzická vrstva....41
4.5.3 Formát rámce....42
4.6 Výběr komunikační sběrnice....43
4.7 Návrh komunikačního protokolu....45
4.7.1 HDLC protokol....45
4.7.2 Modifikace HDLC protokolu....47

4.7.3 Návrh struktury přenášených dat....49
5 Návrh softwaru....51
5.1 Hlavní řídící jednotka....51
5.1.1 Struktura softwaru....51
5.1.2 Modul komunikace....52
5.1.3 Regulace teploty v kupé....61
5.2 Jednotka v kupé....64
5.2.1 Struktura softwaru....64
5.2.2 Modul komunikace....65
6 Testování....69
7 Závěr....72
8 Seznam použité literatury....73
9 Přílohy....75

Seznam obrázků

Obr. 2.1: Blokové schéma proudění vzduchu ve voze....4

Obr. 2.2: Návrh řešení....5

Obr. 2.3: Blokové schéma chladicího okruhu klimatizace....6

Obr. 2.4: Návrh modulárního řešení....9

Obr. 2.5: Návrh monolitického řešení....10

Obr. 3.1: Blokové schéma hlavní řídicí jednotky....16

Obr. 3.2: EMC filtr napájení....20

Obr. 3.3: Spínaný stabilizovaný zdroj 5V....20

Obr. 3.4: Lineární stabilizovaný zdroj 3.3V....21

Obr. 3.5: Galvanicky oddělený zdroj pro RS485....21

Obr. 3.6: Lineární stabilizovaný zdroj 12V....22

Obr. 3.7: Reference 1.5V pro analogová měření....23

Obr. 3.8: Logický vstup....23

Obr. 3.9: Budič logických výstupů....24

Obr. 3.10: Galvanické oddělení pro budič logických výstupů....25

Obr. 3.11: Zpětná vazba od budiče logických výstupů....25

Obr. 3.12: Analogové měření proudové smyčky 4 - 20mA....26

Obr. 3.13: Analogové měření teploty z čidel NTC1000....27

Obr. 3.14: Analogové měření napětí....28

Obr. 3.15: EMC filtr pro proudovou smyčku....28

Obr. 3.16: EMC filtr pro čidla NTC....28

Obr. 3.17: EMC filtr měřeného napětí....29

Obr. 3.18: Analogový výstup 0 - 10V....29

Obr. 3.19: Budič RS485....30

Obr. 3.20: Blokové schéma jednotky v kupé....31

Obr. 4.1: Schéma budiče LIN[9].....35

Obr. 4.2: Formát rámce LIN[9].....36

Obr. 4.3: Fyzická vrstva CAN [10].....38

Obr. 4.4: Průběhy signálů CAN [10].....39

Obr. 4.5: Dvouvodičová verze RS485 [12].....41

Obr. 4.6: Čtyřvodičová verze RS485 [12].....42

Obr. 4.7: Formát rámce HDLC protokolu [14].....45

Obr. 4.8: Návrh komunikačního protokolu.....47

Obr. 5.1: Blokové schéma softwaru pro hlavní řídicí jednotku.....51

Obr. 5.2: Blokové schéma software pro komunikaci na straně masteru.....52

Obr. 5.3: Stavový automat pro příjem datagramů.....54

Obr. 5.4: Stavový automat master.....57

Obr. 5.5: Dvoustavová regulace [17].....61

Obr. 5.6: Blokové schéma SW pro regulaci teploty v kupé.....62

Obr. 5.7: Blokové schéma software pro jednotku v kupé.....64

Obr. 5.8: Blokové schéma software pro komunikaci na straně slave.....65

Obr. 5.9: Stavový automat slave.....66

Obr. 9.1: Konečná podoba hlavní jednotky.....75

Obr. 9.2: Kontejner s kompresorem.....76

Obr. 9.3: Kontejner s výparníkem a topnými tělesy.....77

Seznam tabulek

Tab. 2.1: Napětí v průběžném vedení....3

Tab. 3.1: Logické vstupy....11

Tab. 3.2: Logické výstupy....13

Tab. 3.3: Analogové vstupy....13

Tab. 3.4: Analogové výstupy....14

Tab. 4.1: Vlastnosti sběrnic....43

Tab. 4.2: Formát dat - odesílání....49

Tab. 4.3: Formát dat - příjem....50

1 Úvod

Cestování vlakem nabízí rychlý a bezpečný způsob dopravy. V posledních letech zájem cestujících roste zejména na dálkových spojích. Po vstupu konkurence na železnici v roce 2011 začali zákazníci daleko více vnímat nejen ceny jízdného, ale také poskytovaný komfort. Nabízené služby a pohodlí mají pro cestující velký vliv na výběr svého oblíbeného dopravce. Neustálé zkvalitňování poskytovaných služeb je jedinou cestou k úspěchu v konkurenčním boji společností na poli vlakové dopravy. Zajištění požadovaného komfortu v sobě také zahrnuje mimo jiné i tepelnou pohodu cestujících v kupé železničního vozu.

Tato diplomová práce se zabývá návrhem a realizací řídicího systému klimatizace, určené pro železniční vozy řady ABmz. Práce je rozdělena do několika na sebe navzájem navazujících kapitol. Na začátku se věnuje rozboru současného stavu klimatizace ve vozech a důvodům, proč bylo třeba navrhnout novější a modernější řešení. Z této základní analýzy shromažďuje souhrn požadavků na navrhovaný systém. Na jejich základě navrhuje a diskutuje strukturu celého řídicího systému. Dále se věnuje podrobnému popisu požadované funkčnosti jednotlivých celků, ze kterého vyplývá konkrétní elektrický návrh hardwarových komponent. Tímto způsobem vzniká mimo jiné také požadavek na schopnost komunikace mezi těmito komponentami. Velkou pozornost věnuje práce výběru komunikační linky na základě specifikovaných požadavků, s následným návrhem vhodného komunikačního protokolu, včetně jeho softwarové implementace a testování. Poslední část je věnována návrhu řízení tepelné pohody v rámci jednotlivých kupé. V závěru hodnotí práce použité postupy při návrhu a realizaci, včetně námětů na jejich budoucí zlepšení.

Obsah diplomové práce se zabývá reálným projektem, který nastoupí do ostrého provozu. Proto je při jakémkoli návrhu kladen důraz na odolnost a spolehlivost řešení. Případné vzniklé chyby by byly potrestány náročným a hlavně nákladným procesem jejich nápravy.

Na vývoji klimatizace jsem spolupracoval v rámci projektu společnosti Poll s.r.o., který vznikl na objednávku jednoho z konkurentů státního dopravce.

2 Popis řešeného problému

2.1 Princip klimatizace

Klimatizace (označované zkratkou AC z anglického air conditioning) je zařízení pro úpravu vzduchu v celých budovách či jednotlivých místnostech a v dopravních prostředcích. Klimatizace má za úkol úpravu teploty vzduchu, jeho filtraci a všeobecně udržování stálých podmínek v požadovaném prostoru, bez ohledu na venkovní prostředí. Klimatizační zařízení pro vytvoření tepelné pohody v požadovaném prostoru využívá několika fyzikálních principů. Základem klimatizačního zařízení je kompresorový chladicí okruh. Princip chladícího cyklu [1]:

- Ve vnitřní jednotce (ve výparníku), přes kterou proudí ochlazovaný vzduch dochází k odpařování chladiva (změna skupenství chladiva z kapalného na plynné). Při odpařování dochází k odebírání tepla z klimatizovaného prostoru a cirkulující vzduch je ochlazen.
- V kompresoru dochází ke stlačení chladiva (zvýší se teplota a tlak chladiva).
- Ve venkovní jednotce (v kondenzátoru) dochází opět ke změně skupenství chladiva (zkapalnění chladiva), při kterém teplo odebrané z klimatizovaného prostoru je odváděno kondenzátorem do okolního, venkovního prostoru.
- Zkapalněné chladivo má za kondenzátorem vysoký tlak, který je snížen kapilárou nebo expanzním ventilem na požadovaný vypařovací tlak a celý cyklus se opakuje.

Klimatizace umožňuje pracovat i v opačném, vytápěcím režimu. Z jednotky se stane tepelné čerpadlo, kdy se odpadní teplo přivádí zpět do prostoru. Z kondenzátoru se stává výparník a z výparníku kondenzátor. Vnější jednotka je tak ochlazována a vnitřní naopak topí. Tohoto způsobu ohřevu vzduchu v místnosti se však využívá pouze příležitostně v období podzimu a velmi mírné zimy (při teplotách 4 – 13 °C) [2]. Koncepce klimatizace v železničním voze ale tuto možnost neuvažuje. Ohřev vzduchu zde probíhá pomocí topných těles.

2.2 Současný stav

Klimatizace ve vozech ABmz má nahradit stávající systém, který ve vozech již existuje. Výměna klimatizačního agregátu má několik důvodů:

- Současný systém klimatizace je na konci své životnosti
- Vysoká poruchovost stávajícího zařízení
- Obtížná údržba z důvodu stáří komponent a shánění náhradních dílů
- Kompresor klimatizace je určen pro vysoké napájecí napětí (snaha nahradit kompresorem na nízké napětí)

Kompresor i topná tělesa jsou napájena přímo z průběžného vedení napájení topení. Napětí v průběžném vedení je závislé na příslušné trakční soustavě, viz. tab. 2.1.

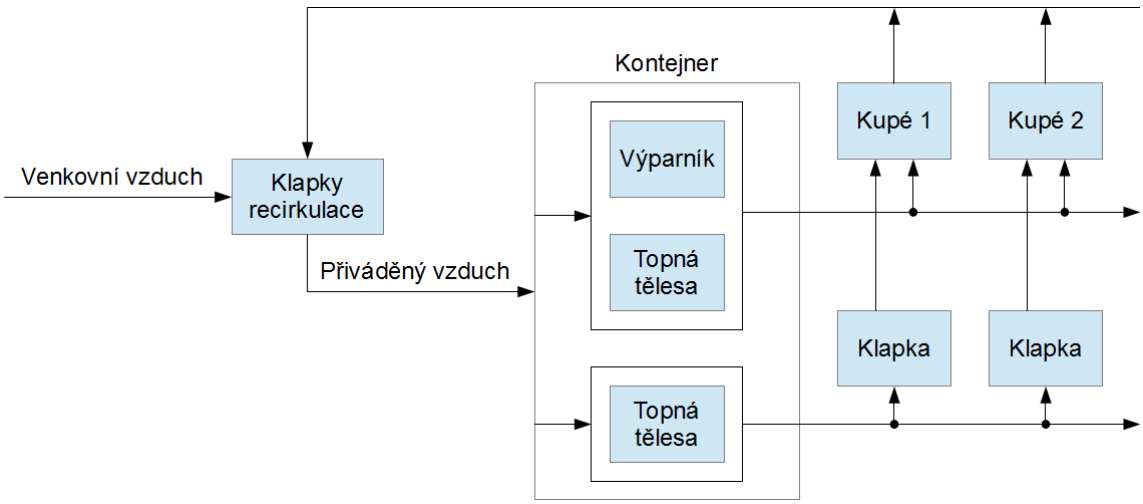
Trakční soustava	Průběžné vedení	Frekvence
1,5 kV DC	1000 – 1800 V	
3 kV DC	2000 – 4000 V	
25 kV AC, 50Hz	1140 – 1760 V ef.	48 – 51 Hz
15 kV AC, 16 2/3 Hz	735 – 1100 V ef.	15 – 17,5 Hz

Tab. 2.1: Napětí v průběžném vedení

Pokud vlak jede v úseku 25 kV/50Hz nebo 15 kV/16 2/3Hz, je napětí sníženo pomocí transformátoru v lokomotivě, bez následného usměrnění. Motor kompresoru je totiž konstruován jako univerzální pro napájení stejnosměrným i střídavým proudem. Topná tělesa lze samozřejmě také napájet stejnosměrným i střídavým proudem.

Na podvozku železničního vozu se nacházejí dva kontejnery. V prvním kontejneru je umístěn kompresor spolu s kondenzátorem a ventilátory na odvod tepla z okruhu chladiva do okolí. Ve druhém kontejneru je výparník klimatizace, topná tělesa pro ohřívání a ventilátor, který vhání ochlazený, nebo ohřátý vzduch do prostorů vozu prostřednictvím dvou vzduchových kanálů. V prvním kanále se nachází výparník a topná tělesa. Výstupní vzduch jde rovnou do jednotlivých kupé. Ve druhém kanále jsou pouze topná tělesa. Tento kanál si ale může každé kupé otevírat nebo zavírat

prostřednictvím elektromagnetické klapky (má pouze dva stavy – otevřeno, zavřeno). Ve voze existuje ještě jeden kanál, který je určen pro recirkulaci vzduchu zpět do klimatizační jednotky. Je ovládán klapkami s možností nastavení 0%, 50%, nebo 100% recirkulování vzduchu. Obr. 2.1 znázorňuje uspořádání přívodu vzduchu do jednotlivých kupé.



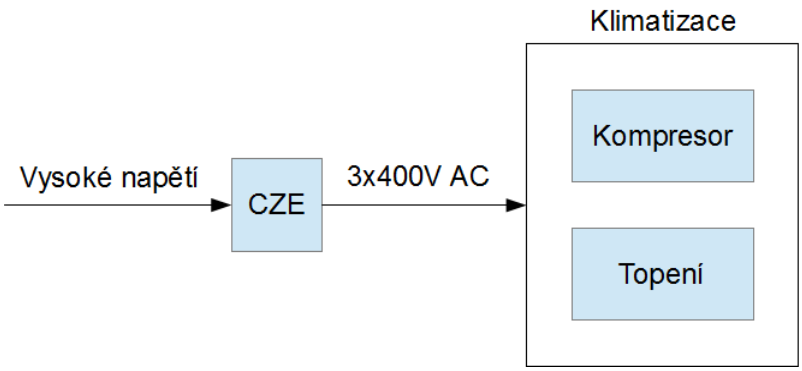
Obr. 2.1: Blokové schéma proudění vzduchu ve voze

Řízení ve stávajícím stavu probíhá tak, že kompresor je většinu času zapnutý (pokud to venkovní teplota vyžaduje). V tepelném kanále jsou zapnuta topná tělesa a každé kupé pomocí klapek řídí výslednou teplotu mícháním studeného vzduchu od kompresoru a teplého vzduchu od topení. Spínání klapek v kupé je realizováno rtuťovým spínačem. Pokud je teplota nižší než požadovaná, klapka je otevřená a přivádí se do kupé teplý vzduch (společně se studeným). Jak teplota roste, roste také hladina rtuti ve spínači až do chvíle, kdy elektricky spojí kontakty, které zajistí zavření klapky.

Tento způsob řízení má jeden velký nedostatek. V jednu chvíli běží kompresor a topná tělesa zároveň i v případě, že to nemusí situace vůbec vyžadovat. To má za následek vysokou spotřebu elektrické energie. Takové řízení klimatizace je tedy velmi neefektivní.

2.3 Návrh řešení

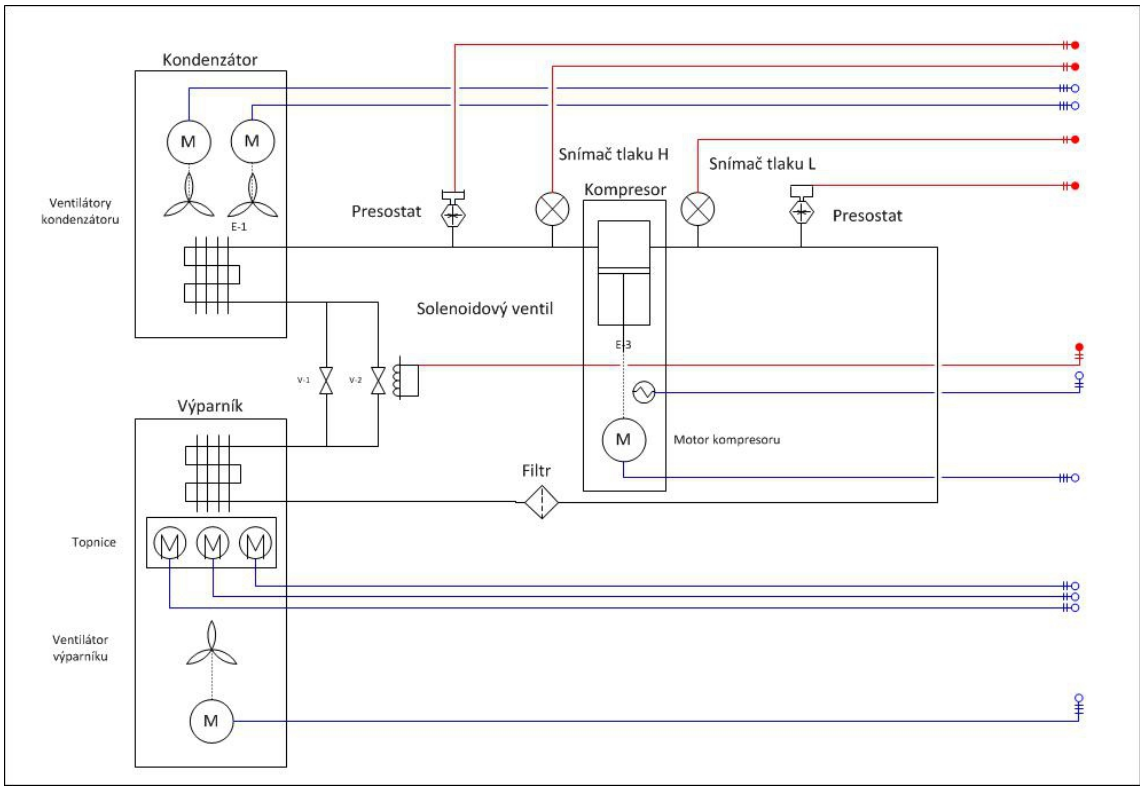
Navrhované řešení mění celý kontejner s kompresorem a samozřejmě veškerou elektroniku, která se stará o chod celého agregátu. Kompresor již není napájen zdrojem vysokého napětí, ale jednotka CZE (Centrální zdroj energie) mění vysoké napětí na třífázové 3x400V AC/50Hz. Odtud je napájen jak kompresor, tak topná tělesa. Jednotka CZE je ve vozech ABmz již nainstalována. Tuto popisovanou změnu ilustruje obr. 2.2.



Obr. 2.2: Návrh řešení

Co zůstává zachováno je kontejner, kde se nachází výparník a topná tělesa. Stejně tak zůstává původní princip rozvodu vzduchu v kanálech a funkce elektromagnetických klapek v jednotlivých kupé.

V této chvíli můžeme definovat všechny prvky, které bude muset nová elektronika schopna obsluhovat. Pro lepší vysvětlení je na obr. 2.3 schéma chladicího okruhu klimatizace.



Obr. 2.3: Blokové schéma chladicího okruhu klimatizace

Komponenty obsluhované elektronikou:

- **Stykače kompresoru** – kompresor má dvě vinutí pro 3x400V AC (rozběh a běh) zapojené do hvězdy. Dále má kompresor vytápění oleje a vlastní diagnostickou elektroniku, která řídicí elektronice hlásí stav kompresoru.
- **Stykače ventilátoru kondenzátoru** – dva ventilátory kondenzátoru jsou napájeny 3x400VAC. Vinutí jsou zapojena do hvězdy.
- **Stykače ventilátoru přiváděného vzduchu** – ventilátor je napájen 24V DC a má dvě vinutí (pomalý a rychlý chod).
- **Stykače topných těles** – topná tělesa jsou zapojena do hvězdy a na jejich spínání potřebujeme celkem čtyři stykače. Dva bloky těles jsou v kanálu společně s výparníkem, další dva bloky jsou v topném kanále.

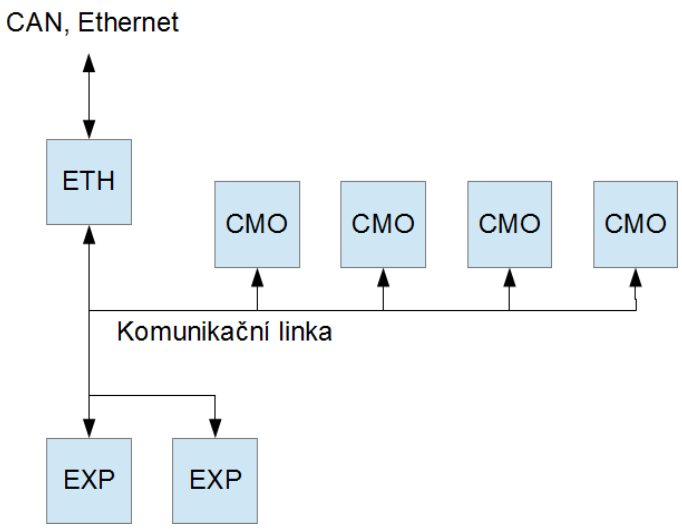
- **Solenoidový ventil** – tímto ventilem se uzavírá okruh chladiva, pokud není kompresor v činnosti. Zavřením ventilu se zabraňuje, aby se zkapalněné chladivo nedostalo zpět do kompresoru, což by mohlo mít za následek jeho zničení.
- **Presostaty** – presostat je zařízení, které kontroluje tlak v okruhu chladiva. Pokud by selhalo řízení elektroniky a tlak chladiva se začal pohybovat v nedovoleném intervalu, presostat rozepne stykač kompresoru, aby nedošlo k jeho zničení. V okruhu jsou tato zařízení dvě. První hlídá překročení maximálního povoleného tlaku za kompresorem (18 bar). Druhé hlídá, aby tlak nepoklesl pod minimální hranici před kompresorem (1 bar). Pokud by byla jedna z podmínek splněna, presostaty okamžitě odstavují kompresor. Elektronika se o tomto zásahu dozví pomocí zpětné vazby od presostatu.
- **Měření tlaků** – měříme tlak v okruhu chladiva před kompresorem a za kompresorem.
- **Měření teplot** – v celém systému měříme několik teplot. Venkovní teplota, teplota nasávaného vzduchu, teplota recirkulovaného vzduchu a teplota v prvním a ve druhém vzduchovém kanále.
- **Termostaty u topných těles** – při překročení maximální teploty v jednom nebo druhém kanále termostaty automaticky vypínají topná tělesa, aby nedošlo k jejich zničení. Elektronika zásah těchto termostatů kontroluje.
- **Termostat u teploty za výparníkem** – hlídá minimální povolenou teplotu vzduchu za výparníkem, kvůli možnosti jeho namrzání.
- **Větrné relé** – dodává do elektroniky informaci o tom, jestli skutečně proudí vzduch v obou kanálech a nedošlo k poškození ventilátoru přiváděného vzduchu.

Dále musíme mít k dispozici elektroniku, která bude v každém kupé měřit skutečnou teplotu a zjišťovat požadovanou teplotu nastavenou cestujícími. Zároveň bude také schopna otevírat a zavírat klapku pro ovládání přiváděného vzduchu z tepelného kanálu do kupé. Regulovaná teplota se může od nominální pohybovat ±2 °C. Nominální teplota se pohybuje v rozsahu 21 – 28 °C a závisí na aktuální venkovní teplotě.

Je tedy zřejmé, že budeme potřebovat dva základní bloky elektroniky. První se bude kompletně starat o řízení klimatizačního agregátu, druhý bude zpracovávat požadavky na jednotlivá kupé. Následující kapitola konzultuje dva základní přístupy pro řešení tohoto problému.

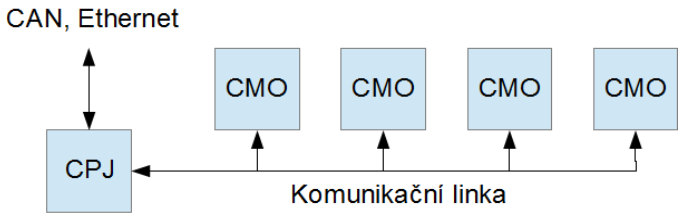
2.3.1 Modulární vs. monolitické řešení

Existují minimálně dvě varianty, jak celý systém navrhnout. První variantou je, že veškeré zpracování informací bude mít na starosti centrální jednotka, ke které budou připojeny jednotlivé jednotky v kupé (na obrázku CMO) a také expandery (EXP). Ty by zajišťovaly spínání stykačů, měření tlaků a teplot na klimatizačním agregátu. Všechny tyto procesy by byly řízeny z jedné centrální jednotky pomocí komunikační linky (na obrázku označeno ETH). Expander by byla univerzální jednotka s určitým počtem digitálních a analogových vstupů a výstupů. Toto řešení má jednu velkou výhodu. V budoucích projektech se může stát, že bude potřeba více, či méně vstupů nebo výstupů. Tento nedostatek by bylo možné lehce vykompenzovat přidáním, nebo ubráním počtu expanderů. Nevýhodou tohoto řešení může být vyšší náročnost na komunikaci a na její spolehlivost. Na obr. 2.4 je blokové schéma tohoto modulárního řešení.



Obr. 2.4: Návrh modulárního řešení

Ve druhé variantě by existovala pouze jedna jednotka (na obrázku CPJ), která nahrazuje funkci expanderů z předchozího obrázku a zároveň implementuje veškerou logiku z hlediska řízení klimatizačního agregátu. Toto řešení je pravděpodobně jednodušší na vývoj hlavně ze strany testování. Jeho nevýhodou ale je, že má pevně daný počet vstupů a výstupů s nemožností jejich jednoduchého rozšíření. Jednotky v kupé (CMO) jsou připojeny přes komunikační linku stejně jako v předchozím případě k centrální jednotce. Blokové schéma této varianty je na obr. 2.5.



Obr. 2.5: Návrh monolitického řešení

Osobně se mi více líbí modulární varianta, protože hlavní jednotka, která by celý agregát řídila (ETH) je již ve firmě vyvinuta a vývoj by se zaměřil na expandery. Ty také nemusí mít tak výkonné procesory protože hlavní logika by byla v jednotce ETH. Složitost návrhu elektroniky a plošného spoje by také určitě byla menší než ve druhém případě. Proto si dovolím tvrdit, že vývoj takových expanderů by byl levnější, než v případě monolitického řešení. Modulární řešení nám také přináší velkou flexibilitu v podobných budoucích projektech. Po zvážení těchto dvou variant vedení společnosti rozhodlo, že se vydáme cestou monolitického řešení.

3 Návrh hardwaru

Tato kapitola se bude zabývat návrhem hardwaru. Nejprve si shrneme, jaké vstupy a výstupy od jednotek požadujeme a k čemu budou sloužit. Potom bude následovat elektrický návrh, rozdělený do jednotlivých funkčních bloků.

3.1 Požadavky na hardware

Abychom byli schopni ovládat celý klimatizační agregát, musí řídící jednotka obsahovat kromě procesorové jednotky také vstupy, výstupy (jak digitální tak analogové) a komunikační rozhraní. V první tabulce 3.1 je shrnutí požadovaných logických vstupů včetně popisu jejich funkce.

Logické vstupy
Podpěťová ochrana – Ubat > 20,5V
Dohled stykače ventilátoru př. vzduchu
Síť 400V OK
Větrné relé ventilátoru
Teplota za výparníkem (> 5°C)
Teplota horního kanálu (< 120°C)
Teplota dolního kanálu (< 125°C)
Běh kompresoru
Ochrany kompresoru
Vstupní tlak kompresoru (> Pmin)
Vstupní tlak kompresoru (< Pmax)
Dohled stykačů ventilátorů kondenzátoru
Dohled stykačů topných větví (STT:1,2) - horní kanál
Dohled stykačů topných větví (STT:3,4) - dolní kanál
Celkem: 14

Tab. 3.1: Logické vstupy

- **Podpěťová ochrana (20,5V)** - při poklesu napájení pod tuto mez se klimatizace odstavuje.

- **Dohled stykače ventilátoru přiváděného vzduchu** - Pokud by nefungoval stykač ventilátoru, mohl by začít namrzat výparník, nebo by mohlo dojít k poškození topných těles.
- **Síť 400V AC OK** - při poklesu, nebo výpadku 3x400V AC nemůže fungovat kompresor, ani topná tělesa.
- **Větrné relé ventilátoru** - hlásí poškozený ventilátor přiváděného vzduchu (následky stejné, jako u „Dohled jističe ventilátoru přiváděného vzduchu“).
- **Teplota za výparníkem (> 5°C)** - při poklesu pod tuto hranici začne namrzat výparník.
- **Teplota horního a dolního kanálu** - při překročení této hodnoty by mohlo dojít k poškození topných těles.
- **Ochrany kompresoru** - Zpětná vazba od elektroniky kompresoru (je jeho součástí).
- **Vstupní tlak kompresoru (> Pmin)** - zpětné hlášení od presostatu nízkého tlaku (při poklesu tlaku pod dovolenou mez dojde okamžitě k odstavení kompresoru, bez zásahu řídící jednotky).
- **Výstupní tlak kompresoru (< Pmax)** - zpětné hlášení od presostatu vysokého tlaku (při vzrůstu tlaku nad dovolenou mez dojde okamžitě k odstavení kompresoru, bez zásahu řídící jednotky).
- **Dohled stykačů ventilátorů kondenzátoru** - při výpadku funkce stykače ventilátorů kondenzátoru by při zapnutém kompresoru začal v okruhu chladiva růst tlak nad přípustnou mez (zasáhl by presostat a kompresor odstavil).

V následující tabulce je výčet požadovaných logických výstupů a komponent, které tyto výstupy ovládají (tab. 3.2).

Logické výstupy
Stykač vytápění nástupních prostorů
Stykač ventilátoru přiváděného vzduchu - I.stupeň
Stykač ventilátoru přiváděného vzduchu - II.stupeň
Signálka: provoz klimatizace
Signálka: porucha klimatizace
Signálka: chlazení v provozu
Signálka: porucha chlazení
Ventil výparníku (solenoid)
Stykač napájení pomocných obvodů kompresoru (MP10)
Klapka recirkulace 1
Klapka recirkulace 2
Stykač rozběhu kompresoru
Stykač běhu kompresoru
Stykač topení - horní kanál (1.větev)
Stykač topení - horní kanál (2.větev)
Stykač topení - dolní kanál (3.větev)
Stykač topení - dolní kanál (4.větev)
Stykač ventilátoru kondenzátoru 1
Stykač ventilátoru kondenzátoru 2
Celkem: 19

Tab. 3.2: Logické výstupy

Všechny prvky klimatizace uvedené v tabulce potřebují pro svoji činnost (sepnutý, nebo aktivní stav) výstup z jednotky 24V DC.

Následuje výčet analogových vstupů. Jedná se o měření teploty na různých místech celého klimatizačního agregátu a měření tlaků v okruhu chladiva (tab. 3.3).

Analogové vstupy
Čidlo teploty - sací ventilátor
Čidlo teploty - horní kanál
Čidlo teploty - dolní kanál
Čidlo teploty - odsávání, recirkulace
Čidlo teploty - venkovní pod vozem
Snímač tlaku - vstup kompresoru
Snímač tlaku - výstup kompresoru
Celkem: 7

Tab. 3.3: Analogové vstupy

Teplota je měřena NTC termistory napájené zdrojem konstantního proudu. Snímače tlaků používají standardní proudovou smyčku 4 – 20 mA (odtud jsou i napájeny).

V budoucnu se by se měl uskutečnit podobný projekt klimatizace pro jiného zákazníka, kde se recirkulační klapky neovládají digitálním signálem, ale analogovým s požadovaným výstupem 0 – 10V. Proto je nutné do návrhu zahrnout i analogové výstupy (tab. 3.4).

Analogové výstupy
Klapka recirkulace 1 (budoucí projekt)
Klapka recirkulace 2 (budoucí projekt)
Celkem: 2

Tab. 3.4: Analogové výstupy

Mezi další požadavky k hlavní jednotce patří komunikační linka, abychom byli schopni vyměňovat informace s jednotlivými jednotkami v kupé. Podrobnější analýze a výběru komunikační linky se věnuje kapitola 4. Komunikace.

Od jednotky v kupé požadujeme daleko méně vstupů a výstupů, protože v podstatě veškerá část řízení se bude odehrávat v hlavní jednotce. Elektronika v kupé potřebuje dva logické výstupy (24V DC) pro ovládání elektromagnetických klapek ve vzduchovém kanále. Téměř ve všech kupé ve voze bude použitý pouze jeden výstup (klapka pouze na tepelném kanálu). Výjimku tvoří první a poslední kupé, ze kterých se bude ovládat i druhá elektromagnetická klapka, která se nachází na WC. Jednotka by také měla být schopna umět tzv. mód autoregulace. Přechod do autoregulace je podmíněn výpadkem komunikace. V tomto módu převezme elektronika v kupé řízení klapek bez zásahu hlavní řídící jednotky. Dále musí být schopna měřit teplotu a číst stav otočného přepínače pro zjištění požadované teploty. Tuto teplotu si určují cestující v kupé. Ve voze se bude nacházet deset těchto jednotek.

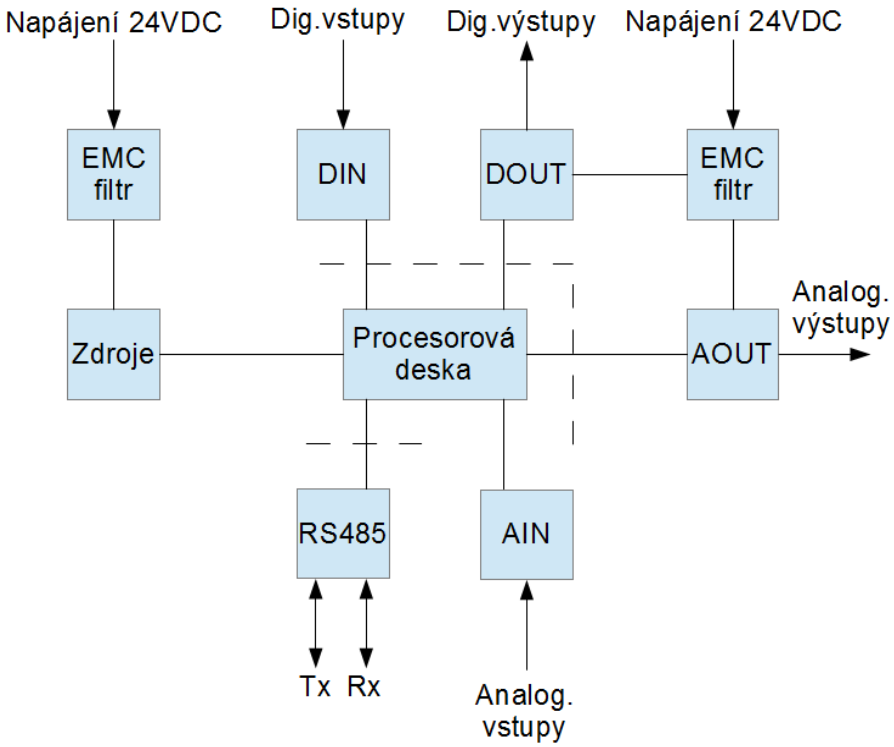
Všechny vstupy a výstupy z jednotek (hlavní, v kupé) musí splňovat elektromagnetickou kompatibilitu (EMC). Elektromagnetická kompatibilita EMC je definována jako schopnost zařízení, systému či přístroje vykazovat správnou činnost i v prostředí, v němž působí jiné zdroje elektromagnetických signálů (přírodní či umělé) a naopak svou vlastní elektromagnetickou činností nepřípustně neovlivňovat své okolí, tj. nevyzařovat signály, jež by byly rušivé pro jiná zařízení [3]. Požadavky na elektromagnetickou kompatibilitu pro náš hardware je definován normou ČSN EN 50121, část 3.2 Drážní vozidla – zařízení.

Tímto bychom měli definovány požadavky na hlavní řídící jednotku i na jednotku v kupé. Následující kapitola se věnuje samotnému návrhu hardware.

3.2 Hlavní řídicí jednotka

3.2.1 Struktura řídicí jednotky

Na obr. 3.1 je blokové schéma hlavní řídicí jednotky. Jednotlivé bloky odpovídají požadavkům specifikovaných v kapitole 3.1.



Obr. 3.1: Blokové schéma hlavní řídicí jednotky

Jednotka je napájena ze dvou větví 24V DC. První větev napájí procesorovou desku společně s analogovými vstupy. Druhá větev napájí digitální a analogové výstupy. Důvodem tohoto řešení je, že první větev je vedena z rozvaděče vozu, kde má obsluha možnost manuálně zapnout nebo vypnout napájecí napětí. Zatímco druhá je přivedena přímo od CZE (centrálního zdroje energie). To může mít za následek rozdílný potenciál v těchto větvích. Z tohoto důvodu jsem se rozhodl pro galvanické oddělení jednotlivých částí řídící jednotky, aby nemohlo docházet k uzavíraní proudové smyčky přes tuto jednotku. Galvanické oddělení znázorňuje na obr. 3.1 přerušovaná linka. Jediný blok, který není takto oddělen jsou analogové vstupy. To nám ale nevadí, protože

čidla budou připojena pouze k této hlavní jednotce a budou odtud i napájena. Tím pádem nemůže dojít k nechtěnému uzavření proudové smyčky přes tyto čidla. Nutno podotknout, že na obrázku je již znázorněna komunikační linka RS485. Vzhledem k tomu, že jsem měl na starosti především komunikaci, je výběr komunikační linky detailněji popsán v kapitole 4. Komunikace. Je tedy zřejmé, že návrh hardware s návrhem komunikace probíhal paralelně.

V této práci navrhuji elektronická schémata všech bloků jednotky (říkejme jí podkladová deska), kromě procesorové desky.

3.2.2 Procesorová deska

Protože ve firmě již mají vyvinutou univerzální procesorovou desku, rozhodl jsem se ji použít z důvodu snížení vývojových nákladů. Všeobecně je tato deska s úspěchem používaná ve více projektech. Jednotka má na konektory vyvedeny GPIO piny, AD převodníky, komunikační linky (CAN, UART) a napájení. Na desce se z periférií nachází obvod RTC (Real-time clock), slot na microSD kartu a Ethernet. Použité procesory:

Procesor TMS320F28335

- DSP (Digital signal processor)
- 512kB FLASH
- 68kB RAM
- 150MHz
- 12bit AD převodníky
- 88 GPIO

Tento procesor je jádrem celého zařízení. Běží zde hlavní program, ovládá vstupy a výstupy, AD převodníky, komunikaci (CAN, UART), PWM, RTC (Real-time clock).

Procesor LM3S8970

- ARM Cortex-M3
- 256kB FLASH
- 64kB RAM
- 50MHz
- Ethernet 10/100Mbps včetně řadiče fyzické vrstvy

Hlavním důvodem tohoto procesoru na desce je brána mezi Ethernetem a UART, protože procesor DSP řadičem Ethernetu nedisponuje. Do procesoru DSP jsou tedy data posílána z procesoru ARM přes UART. Má také na starosti obsluhu microSD karty.

3.2.3 Návrh hardwaru

Při elektrickém návrhu jednotlivých bloků je třeba dbát na dostatečné proudové a napěťové rezervy, EMC kompatibilitu a také na minimální a maximální dovolenou teplotu součástek za provozu (-25°C až +70°C) .

Návrh schémat je vytvořen v softwaru PADS Logic od společnosti Mentor Graphics.

EMC filtry

Na podkladové desce se nacházejí dva EMC filtry napájení. První slouží k filtraci napájení procesorové desky, napájení optočlenů digitálních a analogových výstupů, napájení operačních zesilovačů u analogových vstupů a také napájení pro galvanicky oddělený zdroj komunikační linky. Druhý filtruje napájení komponent, které jsou galvanicky odděleny od jádra jednotky tj. budiče digitálních výstupů, analogový výstup a optočleny digitálních vstupů.

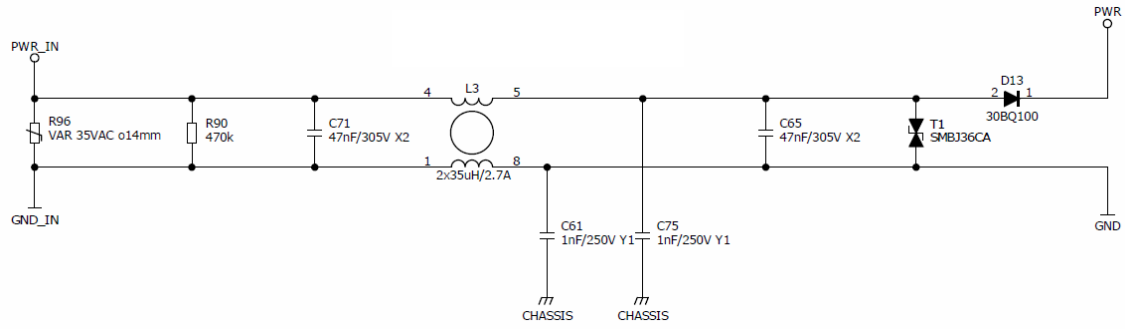
Jedním ze základních odrušovacích prostředků EMC filtru je tlumivka. Pro potlačení nesymetrického rušení, kdy se rušivý proud uzavírá ve stejném směru propojovacími vodiči a vrací se zpět zemí nebo kostrou, lze s výhodou použít tzv. proudově kompenzované tlumivky. Na jednom jádru jsou navinuta vinutí pro všechny propojovací vodiče tak, aby se magnetický tok buzený pracovními proudy vzájemně kompenzoval.

Druhým základním odrušovacím prostředkem je kondenzátor. Pokud má omezovat vysokofrekvenční rušení, a působit tedy jako dolnofrekvenční propust, musí být zapojen příčně, a to pro omezení symetrického rušení mezi každou dvojici propojovacích vodičů, pro omezení nesymetrického rušení mezi každý vodič a zem či kostru. Pro nízké kmitočty je reaktance kondenzátoru velká, a užitečný nízkofrekvenční signál tak neovlivňuje. Naopak vysokofrekvenční rušivý signál je malou reaktancí kondenzátoru téměř zkratován [4].

Na vstupu je paralelně připojen varistor. Varistor je nelineární polovodičový prvek, jehož odpor závisí na napětí. Po dosažení napětí *Un* prudce poklesne vnitřní odpor. Tím varistor působí jako přepěťová ochrana. Na výstupu je paralelně připojen obousměrný transil. Transil je taktéž polovodičový prvek sloužící k ochraně před

napěťovými špičkami. Po překročení prahového napětí se otevře a napětí omezí na velikost danou prahovým napětím. Tím ochrání připojené obvody před zničením přepětím [5].

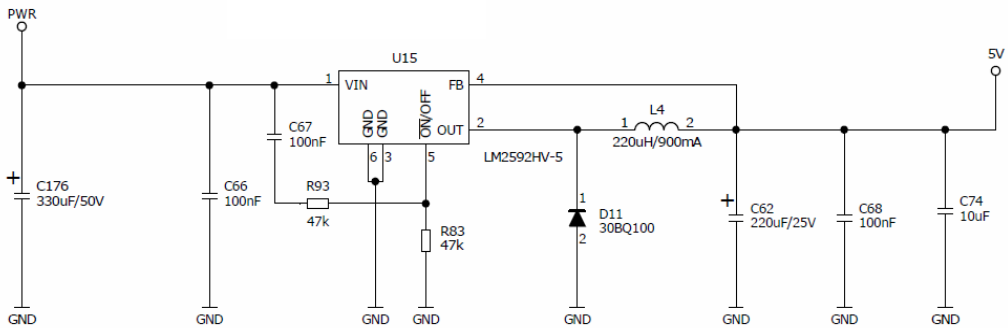
Na obr. 3.2 je schéma EMC filtru. První i druhý filtr je totožný, každý filtruje jen jinou větev napájení tak jak bylo popsáno výše.



Obr. 3.2: EMC filtr napájení

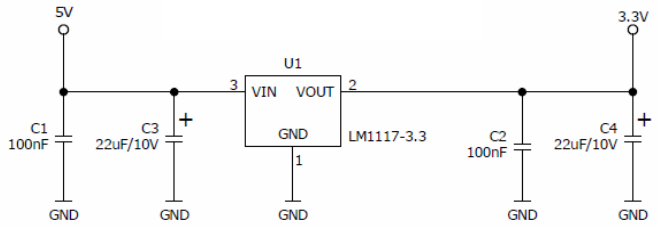
Zdroje

Pro funkčnost jednotky je potřeba několik stabilizovaných zdrojů napětí. Hlavní zdroj napětí 5V je realizován spínaným integrovaným stabilizátorem LM2592HV-5. Maximální vstupní napětí je 60V a dodává do zátěže maximální proud 2A. Tímto zdrojem je napájená celá procesorová deska včetně dalších obvodů (zdroj pro 3,3V, galvanicky oddělený zdroj pro budič linky RS485) na podkladové desce. Na obr. 3.3 je schéma spínaného stabilizovaného zdroje 5V, zapojeného podle doporučené konfigurace od výrobce.



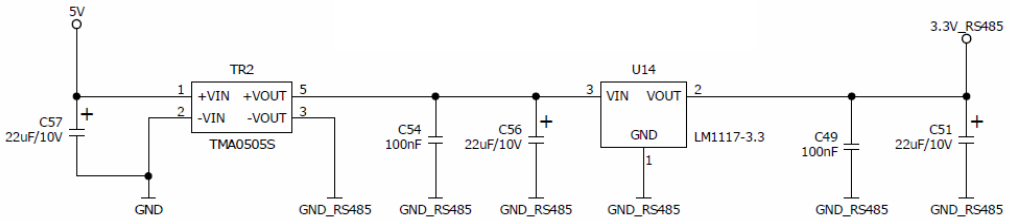
Obr. 3.3: Spínaný stabilizovaný zdroj 5V

Vzhledem k tomu, že většina obvodů na desce (operační zesilovače, hradla, optočleny) jsou napájeny napětím 3,3V, je potřeba další stabilizátor. Ten je již lineární, integrovaný LM1117-3.3 s maximálním proudem 800mA. Na obr. 3.4 je schéma tohoto stabilizátoru v doporučeném zapojení od výrobce.



Obr. 3.4: Lineární stabilizovaný zdroj 3.3V

Protože požadujeme, aby budič komunikační linky RS485 byl galvanicky oddělený, je nutné vytvořit také galvanicky oddělený napájecí zdroj. Zde je použitý spínaný DC-DC, integrovaný, oddělený zdroj, který se dodává jako hotová součástka zalitá v plastovém pouzdru. Parametry zdroje: 5VDC-5VDC, 200mA, 1W. Zde je záměrně použitý zdroj 5VDC-5VDC a nikoliv 5VDC-3.3VDC, protože zkušenosti s tímto zdrojem říkají (reference z oddělení vývoje HW), že nemá příliš kvalitní stabilizovaný výstup. Proto je zde ještě použitý lineární zdroj LM1117-3.3, který stabilizuje napětí na 3,3V. Na obr. 3.5 je schéma celého tohoto zdroje.



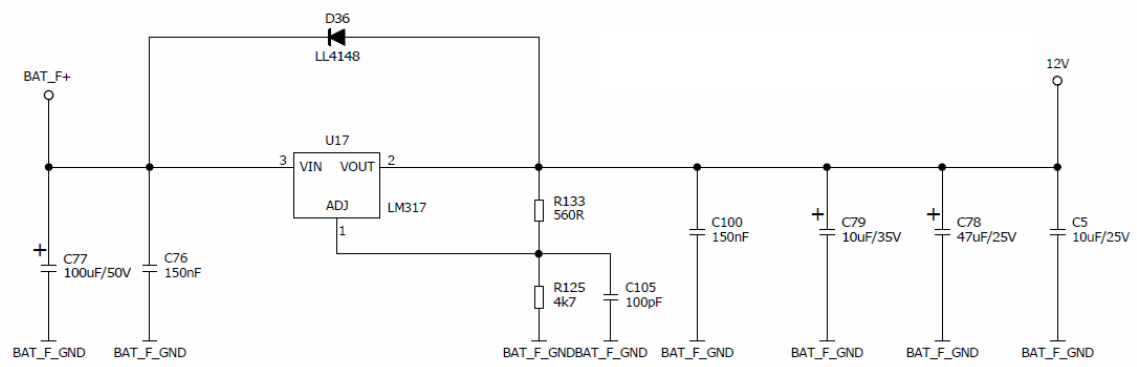
Obr. 3.5: Galvanicky oddělený zdroj pro RS485

Další součástí je pomocný, lineární, stabilizovaný zdroj 12V, který je ale napájen již z druhé větve 24V DC a slouží k napájení obvodů, které jsou připojeny za galvanickým oddělením. Záměrně je zde použitý integrovaný stabilizátor LM317 s nastavitelným výstupním napětím, pro možné budoucí korekce výstupního napětí. Ty

se provádí správným nastavením poměru napěťového děliče, připojeného na řídící elektrodu stabilizátoru podle vztahu [6]:

$$U_{OUT}=1{,}25\cdot\left(1+\frac{R1}{R2}\right)+I_{ADJ}\cdot R1$$

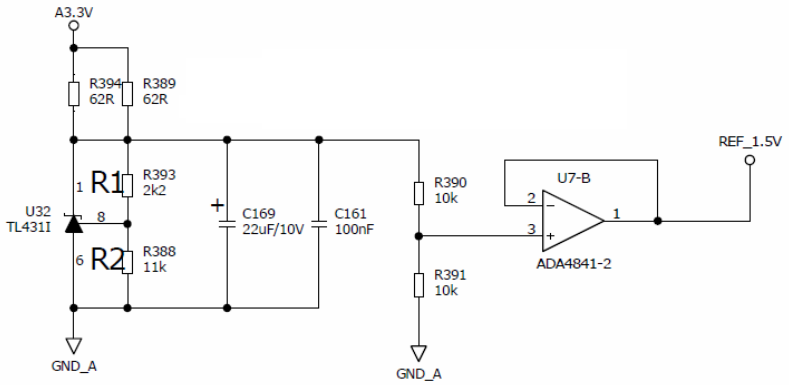
Na obr. 3.6 je schéma lineárního stabilizátoru 12V.



Obr. 3.6: Lineární stabilizovaný zdroj 12V

Poslední zdroj neslouží k napájení obvodů, ale k vytvoření napěťové reference 1,5V pro analogová měření. AD převodník na procesoru DSP (12-bitový) má maximální možné napětí přivedené na jeho vstup 3V (saturace). Pokud budeme potřebovat měřit symetrický signál, musíme ho nejprve stejnosměrně posunout právě o 1,5V aby bylo měřené napětí vždy kladné. Reference 1,5V je docíleno pomocí obvodu TL431 se zaručenou teplotní stabilitou a malým výstupním odporem. Principiálně se chová jako zpětnovazební regulátor s vnější napěťovou zpětnou vazbou a s vnitřní referencí 2,5V. Výstupní napětí 3V je nastaveno pomocí rezistorů R1 a R2. To je pak filtrováno kondenzátory a přes dělič napětí (výstup děliče 1,5V) přivedeno na vstup napěťového sledovače pro snížení výstupního odporu celého obvodu.

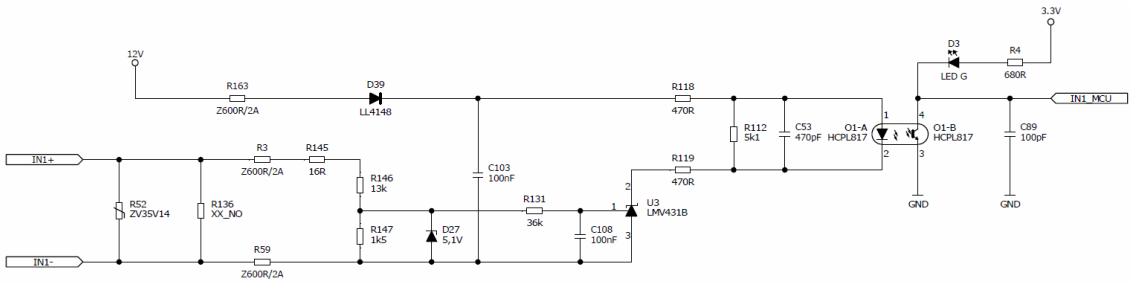
Schéma obvodu pro referenční napětí 1,5V je na obr. 3.7.



Obr. 3.7: Reference 1.5V pro analogová měření

Logické vstupy

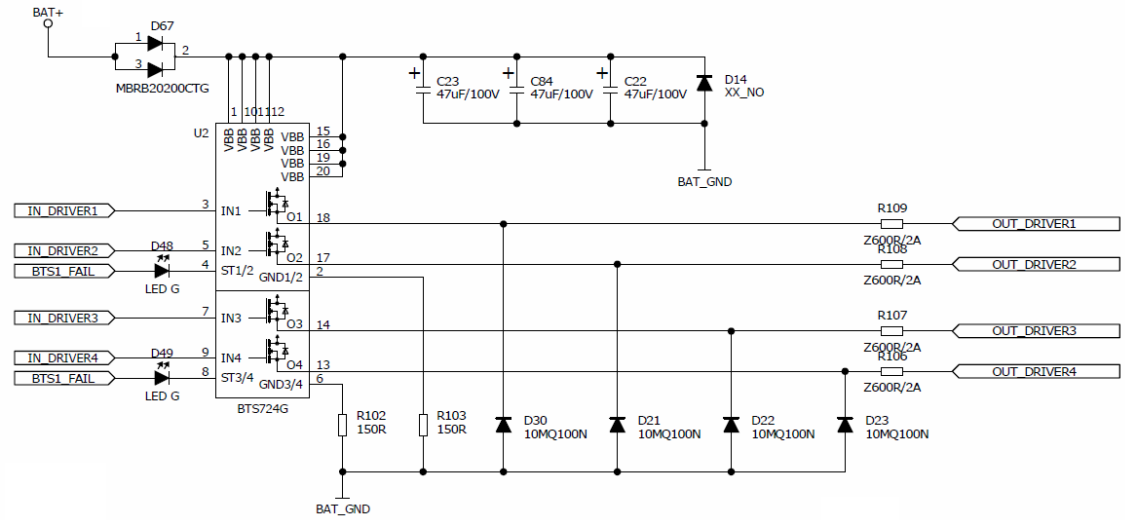
Galvanicky oddělený logický vstup jsem převzal z firmy na radu kolegy z vývoje HW. Je to jejich osvědčené zapojení, které je spolehlivé jak z hlediska funkčnosti, tak z hlediska EMC testů. Na začátku tohoto vstupu je ochranný prvek varistor a dvojice tlumivek. Vstupní napětí je dále přes dělič přivedeno na řídící elektrodu obvodu LM431, který je zde zapojen jako spínač optočlenu. Pokud je na jeho řídící elektrodu přivedeno více jako 2,5V (logická 1 na vstupu), je sepnutý, v opačném případě rozepnutý. Výstup optočlenu je připojen na vstupy procesorové desky. Schéma logického vstupu je na obr. 3.8.



Obr. 3.8: Logický vstup

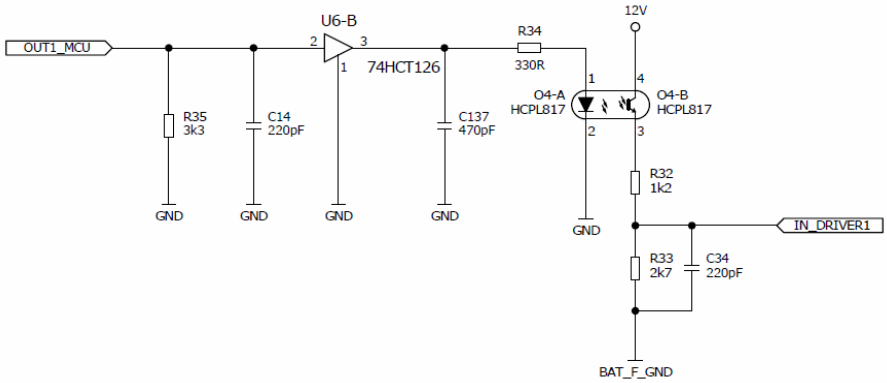
Logické výstupy

Logický výstup je realizován pomocí integrovaného čtyřkanálového budiče BTS724. Tento budič má maximální výstupní proud 10A na kanál a maximální napájecí napětí 36V. Výhoda tohoto řešení je v kompaktnosti a odolnosti. Obvod má zabudované ochrany před zkratem na výstupu, přetížení a překročení maximální teploty. Navíc disponuje zpětnou vazbou, která informuje procesor, zda-li je kanál skutečně sepnutý. Nesepnutí výstupu může mít za příčinu zkrat, nebo překročení dovolené teploty budiče. Na obr. 3.9 je schéma čtyřkanálového budiče.



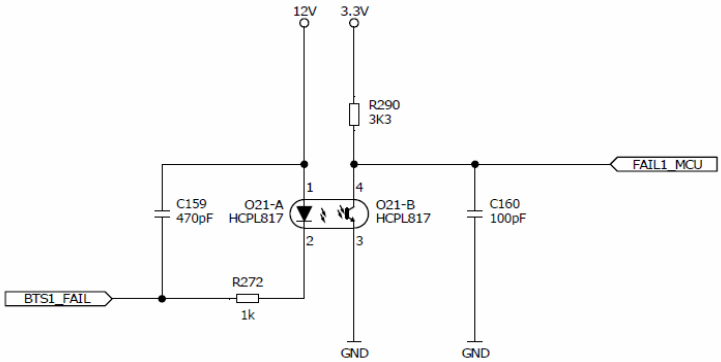
Obr. 3.9: Budič logických výstupů

V návrhu požadujeme, aby byl výstup galvanicky oddělen od obvodů procesorové desky. Toho je docíleno pomocí optočlenů zapojených na obr. 3.10. Hradlo 74HCT126 je zde z důvodu zesílení signálu z procesoru.



Obr. 3.10: Galvanické oddělení pro budič logických výstupů

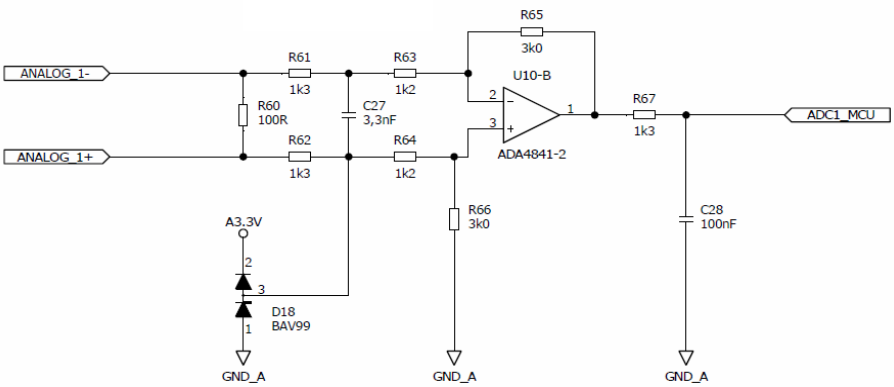
Jak již bylo řečeno, integrovaný budič BTS724 disponuje zpětnou vazbou, která hlásí zpět do procesoru stav (poruchu) svých výstupů. Tuto zpětnou vazbu je opět nutno oddělit optočlenem. Schéma zapojení je na obr. 3.11.



Obr. 3.11: Zpětná vazba od budiče logických výstupů

Analogové vstupy

Sběr analogových veličin je realizován pomocí dvou druhů čidel. Tlaková čidla uzavírají standardní proudovou smyčku 4 – 20 mA. Teplotní čidla jsou typu NTC1000, která mění svůj odpor v závislosti na teplotě. Je nutné napájet je zdrojem konstantního proudu a následně na nich měřit úbytek napětí. Na obr. 3.12 je schéma analogového vstupu pro proudovou smyčku.



Obr. 3.12: Analogové měření proudové smyčky 4 - 20mA

Proud protéká rezistorem R60 o hodnotě 100Ω, na kterém vytváří úbytek napětí podle Ohmova zákona 0,4 – 2V. Toto napětí je přivedeno na vstup diferenčního zesilovače v zapojení s operačním zesilovačem. Má-li diferenční zesilovač skutečně zesilovat jen rozdílové napětí, musí se dodržet následující podmínka[7]:

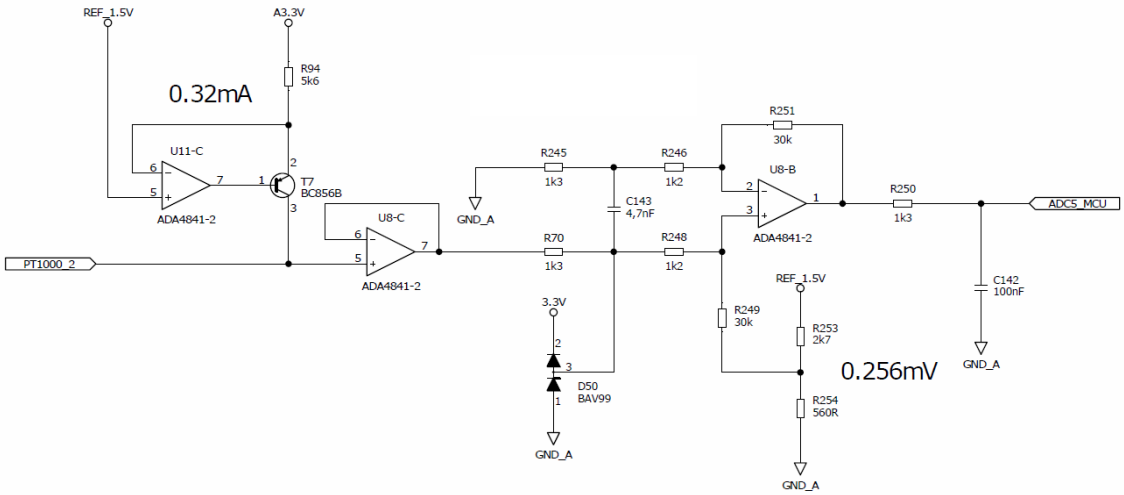
$$R1 = R3\,,\, R2 = R_{zp} \quad ,$$

kde v našem případě *R1 = R61 + R63, R3 = R62 + R64, R2 = R66, Rzp = R65*
Výstupní napětí zesilovače je pak dáno vztahem:

$$U_0 = \frac{R_{zp}}{R_1}(U_2 - U_1)$$

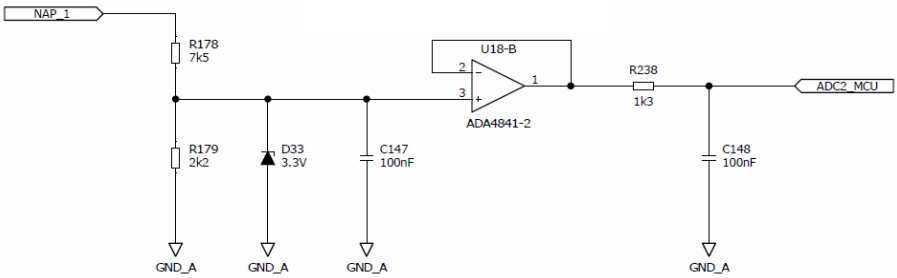
Zesílení našeho diferenčního zesilovače je nastaveno na hodnotu 1,2. Proud 4 – 20 mA vytváří na vstupu AD převodníku procesoru napětí 0,48 – 2,4V.

Na dalším obrázku 3.13 je schéma zapojení pro měření teploty. Podle výrobce je doporučená hodnota proudu čidlem 0,3 mA. To zajišťuje proudový zdroj zapojený pomocí operačního zesilovače a PNP tranzistoru. Princip činnosti tohoto zdroje spočívá v tom, že na invertující vstup zesilovače je přiveden úbytek napětí na rezistoru R94 a na neinvertující vstup je přivedeno referenční napětí 1,5V. Odpor rezistoru je určen tak, aby na něm při proudu 0,32mA vznikal úbytek napětí 3,3V – 1,5V = 1,8V. Snahou operačního zesilovače je tedy otevírat tranzistor takovým způsobem, aby na tomto rezistoru byl úbytek napětí právě 1,8V. Úbytek napětí na termistoru je prostřednictvím sledovače napětí opět přiveden na vstup diferenčního zesilovače. Jeho princip činnosti je stejný, jako u měření proudové smyčky. Zesílení je zde nastaveno na hodnotu 12. Protože pro náš požadovaný rozsah měřených teplot (-40°C až 100°C) má čidlo odpor v rozmezí 800 – 1600Ω (úbytek napětí není nulový u spodní hranice rozsahu teplot), je nutné stejnosměrné posunutí měřeného napětí na rozdílovém zesilovači, abychom využili celé pásmo AD převodníku v procesoru. To zajišťuje dělič napětí R253 a R254.



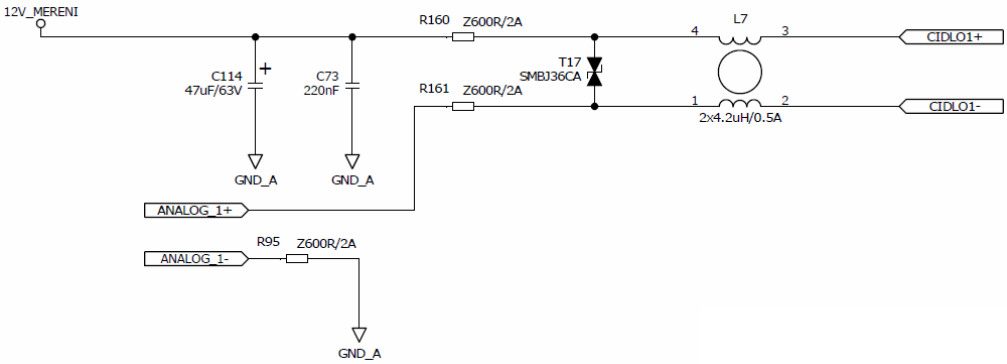
Obr. 3.13: Analogové měření teploty z čidel NTC1000

Pro zajištění požadovaných funkcí klimatizačního agregátu potřebujeme monitorovat také všechna napájecí napětí. (24V, 5V, 3,3V). K tomu slouží zapojení na obr. 3.14. Měřené napětí je přivedeno přes dělič se správně nastaveným poměrem rezistorů na sledovač napětí. Jeho výstup je přímo zapojen na vstup AD převodníku procesoru. Zenerova dioda slouží jako ochrana proti přepětí.



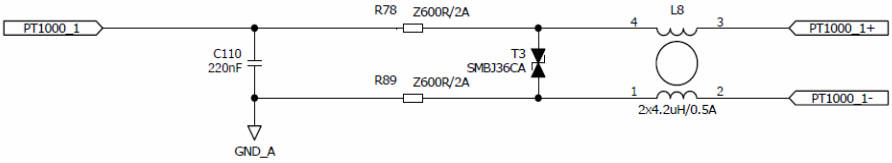
Obr. 3.14: Analogové měření napětí

Všechny analogové vstupy musí mít také EMC filtr. Na obr. 3.15 je schéma EMC filtru pro čidla s proudovou smyčkou 4 – 20 mA. Čidlo je touto smyčkou také zároveň napájeno ze zdroje 12V.



Obr. 3.15: EMC filtr pro proudovou smyčku

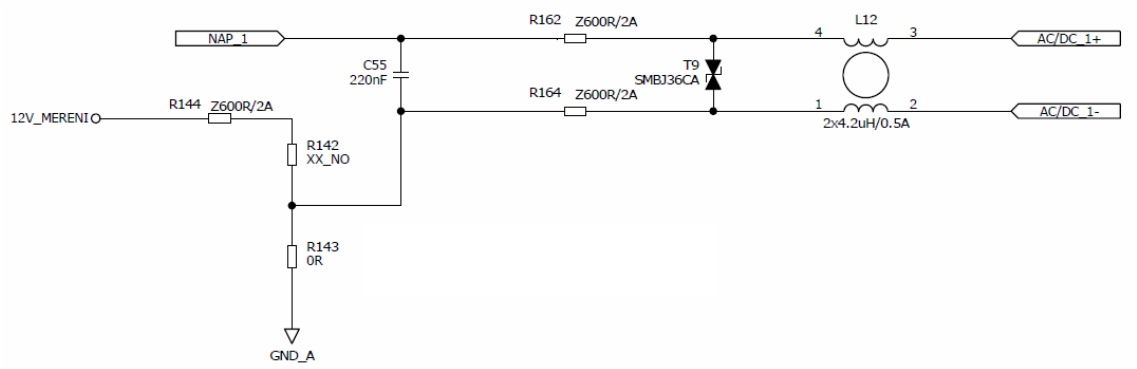
Na obr. 3.16 je situace obdobná, ovšem zde se jedná o EMC filtr pro NTC čidla. Ta jsou napájena zdrojem konstantního proudu z obr. 3.7.



Obr. 3.16: EMC filtr pro čidla NTC

Poslední EMC filtr na obr. 3.17. slouží k filtraci měřeného napětí. Zde existují dvě varianty osazení. Dělič tvořený rezistory R142 a R143 je připraven i pro měření střídavého napětí. Dohled nad sítí 3x400V AC je realizován pomocí transformátorů

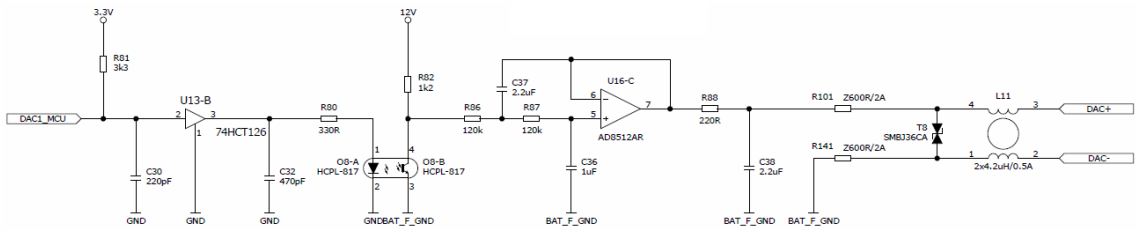
jednotlivých fází. Jejich výstup 12V AC je připojen k tomuto obvodu. Napětí je pak děličem stejnosměrně posunuto, abychom měřili pouze kladné hodnoty.



Obr. 3.17: EMC filtr měřeného napětí

Analogové výstupy

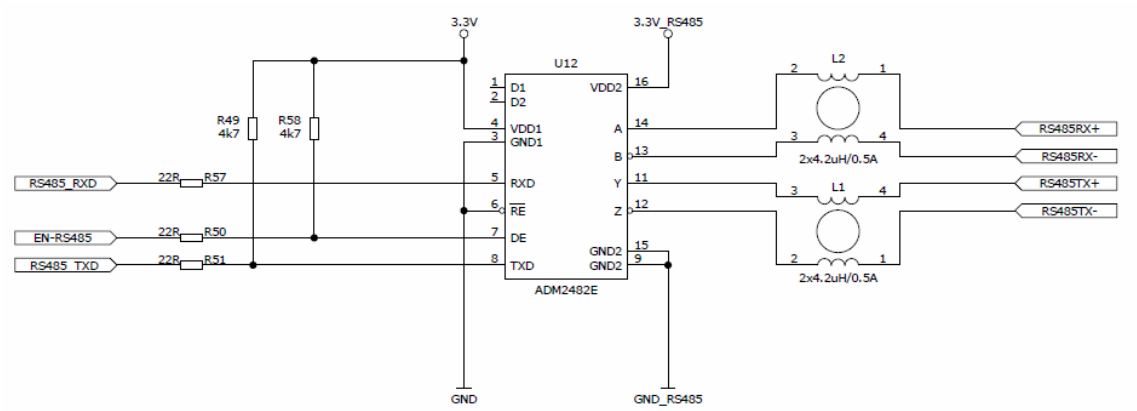
Analogové výstupy slouží pro ovládání recirkulačních klapek (v jiném projektu, kde bude tato jednotka použita). Ty jsou ovládány spojitým stejnosměrným napětím 0 – 10V. Výstupní napětí je tvořeno PWM výstupem z procesoru, kde signál prochází přes zesilující hradlo a optočlen. Za galvanickým oddělením následuje filtr dolní propusti druhého řádu. Mezní frekvence tohoto filtru je nastavena na 1Hz pomocí výpočtu z dokumentu [8]. Výstup je dále filtrován EMC filterm. Schéma zapojení je na obr. 3.18.



Obr. 3.18: Analogový výstup 0 - 10V

RS485

Posledním blokem návrhu je komunikační linka RS485. Zde je použitý budič linky ADM2482E, který je napájen z galvanicky odděleného zdroje 3,3V. Pro datové vodiče není oddělení potřeba, protože je již integrováno v budiči. Na obr. 3.19 je schéma budiče RS485 včetně tlumivek, které tvoří jednoduchý EMC filtr diferenciálních vodičů.



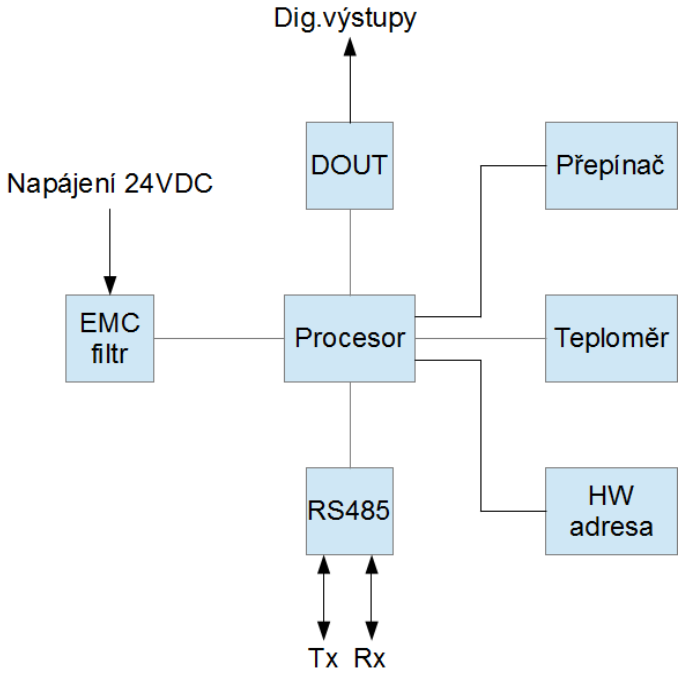
Obr. 3.19: Budič RS485

3.3 Jednotka v kupé

Návrhu hardware jednotky v kupé se věnovali spolupracovníci z našeho týmu. Pro úplnost ale uvádím strukturu jednotky pro objasnění, jakým způsobem tento prvek zapadá do celé koncepce klimatizace.

3.3.1 Struktura jednotky v kupé

Na obr. 3.20 je blokové schéma jednotky v kupé. Cílem návrhu bylo zvolit pokud možno co nejlevnější variantu použitých součástek, kvůli velkému počtu vyrobených kusů (190 jednotek). Jednotka se skládá z následujících funkčních celků:



Obr. 3.20: Blokové schéma jednotky v kupé

- EMC filtr – filtr napájení, který zaručuje EMC kompatibilitu zařízení
- Procesor – MSP430F2132 (8kB FLASH, 512B RAM, 16MHz)
- RS485 – budič sériové linky RS485
- Přepínač – 6-ti polohový přepínač, pro nastavení požadované teploty v kupé

- Teploměr – digitální teplotní čidlo Dallas DS18B20 připojené k procesoru po sběrnici 1-Wire
- HW adresa – nastavitelná hardwarová adresa zařízení pro komunikaci. Provedení je realizováno prostřednictvím jumperů na desce (nastavitelná adresa zařízení je 0 – 32)
- DOUT – digitální výstupy pro ovládání elektromagnetických klapek.

4 Komunikace

Kapitola zabývající se komunikací hlavní řídící jednotky s jednotkami v kupé. Jsou zde popsány tři druhy komunikačních linek, jejich zhodnocení a následný výběr nejvhodnější z nich. Potom následuje návrh vhodného komunikačního protokolu.

4.1 Požadavky na komunikaci

Nedílnou součástí projektu bylo vybrat vhodnou sběrnici pro komunikaci hlavní řídící jednotky s jednotkami v kupé. Rozhodujícími vlastnostmi byla spolehlivost, odolnost proti rušení, cena budičů a cena fyzické vrstvy. Na druhou stranu pro nás nebyla tolik důležitá rychlost komunikace, protože přenášíme především údaje o teplotě, která se z pohledu rychlosti jakýchkoliv linek mění velmi pomalu. Pomocí těchto kritérií jsem se rozhodoval mezi třemi sběrnicemi, LIN, CAN a RS485. Podrobnější rozbor těchto linek a následné zhodnocení jejich vlastností je rozepsáno v následujících kapitolách.

4.2 Sběrnice LIN (Local Interconnect Network)

Sběrnice LIN je sériová asynchronní sběrnice používající ke komunikaci jednovodičové spojení připojených zařízení. Je navržena pro použití v automobilové technice s ohledem na minimální cenové náklady spojené s její aplikací. Současné aplikace vycházejí převážně z oblasti automobilového průmyslu. Jedná se především o ovládání a polohování zrcátek, stahování oken, ovládání zámků dveří a střešního okna, polohování sedadel, ovládání klimatizace, stěračů nebo osvětlení. LIN zde realizuje propojení čidel, ovladačů, akčních členů a indikátorů. To ale neznamená, že tato sběrnice nemůže být použita v jiných oblastí jako je třeba automatizační technika, měřící technika atd. [9].

4.2.1 Vlastnosti sběrnice

- Sériový přenos dat využívající formát UART/RS-232
- Jednovodičová sběrnice
- Komunikace typu master-slave
- Propojení až 17 jednotek (1x master, 16x slave)
- Rychlost komunikace 2400 až 19200 bit/s
- Časová synchronizace bez stabilizované časové základny
- Kontrolní součet dat a detekce chyb
- Volitelná délka datového rámce (2, 4 a 8 byte)

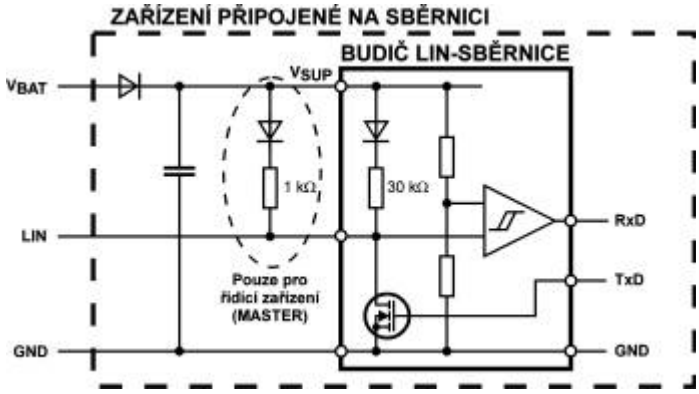
Jedná se o sběrnici typu master-slave (jeden uzel nadřízený a více podřízených), kde jedno řídící zařízení kontroluje komunikaci s jedním nebo více podřízenými zařízeními. Jednotlivá napojení na jednovodičovou sběrnici tvoří drátový součin AND a komunikace probíhá maximální přenosovou rychlostí 19200 bit/s. Ke generování komunikace lze použít hardwarových a softwarových prostředků běžného UART interface, přičemž podřízené jednotky (slave) nepotřebují k činnosti přesný krystalový generátor hodin, ale vystačí jen s RC oscilátorem. Synchronizaci pro komunikaci totiž provádí řídící zařízení (master) na začátku každé komunikace. Výše zmiňované

vlastnosti mají příznivý vliv na cenu komunikačních komponent a umožňují tak snížit cenu i celých jednotlivých zařízení.

4.2.2 Fyzická vrstva

Princip sběrnice LIN spočívá v použití jednoho vodiče pro obousměrnou komunikaci pomocí realizace funkce logického součinu, prostřednictvím spínačů a rezistorů, zapojených na LIN sběrnici v každém připojeném zařízení. Jsou definovány dvě vzájemně komplementární hodnoty stavů na sběrnici a to *dominant* a *recessive*. Spínače při sepnutí spojují sběrnici se zemí, a stačí aby byl sepnut alespoň jeden z nich a sběrnice přejde do stavu dominant, což představuje stav logické nuly. Rezistory zapojené mezi napájecí napětí a sběrnici pak na ní udržují logickou jedničku (stav *recessive*), pokud není žádný spínač sepnutý.

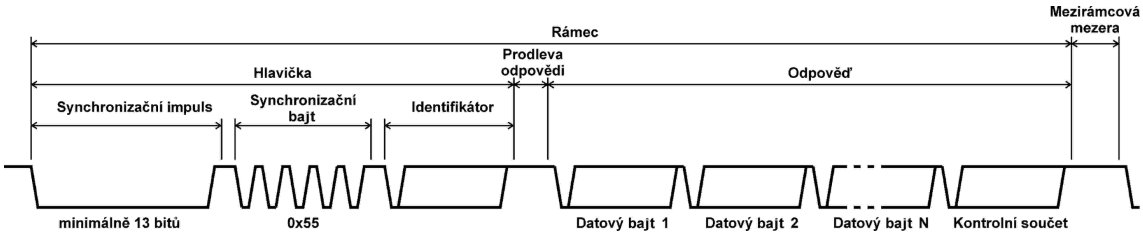
Schéma zapojení budiče sběrnice je na obr. 4.1. Vodiče VBAT a GND slouží k napájení budiče i vlastního zařízení. V případě přerušení napájecího napětí do zařízení připojeného na sběrnici jsou rezistory definující stav *recessive* zapojeny v sérii s ochranou diodou. Ta zabrání nedefinovanému napájení jednotky po vodiči LIN sběrnice. Aby se s počtem připojených budičů razantně neměnila velikost výsledného odporu připojující sběrnici na napájecí napětí, je definováno, že u zařízení typu master, které je na sběrnici vždy jen jedno, je kromě interního rezistoru zapojen navíc externí rezistor o hodnotě 1 kΩ. Maximální délka sběrnice je 40m.



Obr. 4.1: Schéma budiče LIN[9]

4.2.3 Formát rámce

LIN používá jednotný formát rámce zprávy, který slouží k synchronizaci, adresaci uzlů a k výměně dat mezi nimi. Formát rámce zprávy je na obr. 4.2 řídící jednotka (master) začíná komunikaci, určuje přenosovou rychlost a vysílá hlavičku rámce zprávy. Ostatní jednotky, ale i jednotka master mohou vysílat odpověď složenou z datových byte a kontrolního součtu. Hlavička začíná synchronizačním impulsem a následným synchronizačním polem. Toto pole slouží k zasynchronizování podřízených jednotek (slaves) na bitovou rychlost jednotky master. Posledním blokem hlavičky je identifikační pole. Tento identifikátor v sobě nese informaci o odesílateli, příjemci či příjemcích a délku datové části. Je zabezpečen pomocí dvou paritních bitů 6 a 7. Vlastní datová část přenosu je pak zabezpečena invertovanou hodnotou součtu s přenosem přes tyto data. V nové verzi LIN protokolu je pak možnost započítat do tohoto součtu ještě i byte identifikátoru a tím dále zvětšit zabezpečení přenosu proti chybám. Formát LIN rámce je znázorněn na obr. 4.2.



Obr. 4.2: Formát rámce LIN[9]

Jako zdroj signálu pro komunikaci lze použít standardní UART interface s polem jednotlivých byte. Jediná výjimka z 8N1 módu komunikace je synchronizační impuls, který vysílá pouze master. Doba trvání tohoto impulsu odpovídá vyslání minimálně 13 bitů a je většinou generován softwarově. Tento vzor je jednoznačně identifikovatelný protože jeho délka je větší než jakákoliv standardní sekvence na sériovém kanále.

4.3 Sběrnice CAN (Controller Area Network)

CAN je sériová sběrnice vyvinutá firmou Bosch, využívaná nejčastěji pro vnitřní komunikační síť senzorů a funkčních jednotek v automobilu. Původním záměrem byla především úspora kabeláže a zabezpečení přenosu informací mezi snímacími, řídícími a výkonovým prvky. Z této aplikační oblasti se CAN rychle rozšířil také do sféry průmyslové automatizace. Zajišťuje vysokou rychlost přenosu dat, spolehlivost a odolnost při nepříznivých podmínkách (teplota okolí, rušení atd.). Sběrnice CAN původně používala modifikované rozhraní RS485, později bylo definováno normou ISO. Tato norma uvádí specifikaci elektrického rozhraní (fyzická vrstva) a specifikaci datového protokolu (linková vrstva) [10].

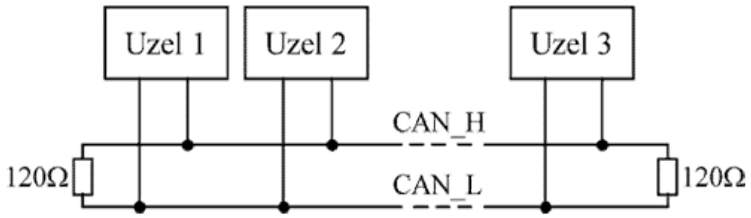
4.4 Vlastnosti sběrnice

- Rychlost komunikace 1Mbit/s při délce sběrnice do 40m
- Rozlišení zpráv identifikátorem CAN 2.0A 11bitů a CAN 2.0B 29bitů
- Selekce přijímaných identifikátorů zpráv
- Prioritní přístup zabezpečující urychlené doručení významných zpráv
- Diagnostika sběrnice: chyba doručení zprávy, chyba CRC, přetečení bufferu
- Stále se rozšiřující součástková základna
- Propojení až 110 jednotek
- Omezený počet dat přenášených v rámci jedné zprávy (0 až 8 Byte)

Komunikace na sběrnici CAN probíhá tak, že každý uzel může za určitých okolností využívat sběrnici pro vysílání svých zpráv. Zpráva vysílaná po sběrnici obsahuje identifikační číslo vysílajícího uzlu. Identifikátor definuje nejen obsah zprávy, ale i prioritu přístupu na sběrnici. Tímto způsobem je možno zaslat zprávu z jednoho uzlu do jiného uzlu nebo několik jiných uzlů současně. Komunikační síť CAN může pracovat jak v režimu multi-master (více nadřízených uzlů), nebo v režimu master-slave (jeden uzel nadřízený a více podřízených uzlů).

4.4.1 Fyzická vrstva

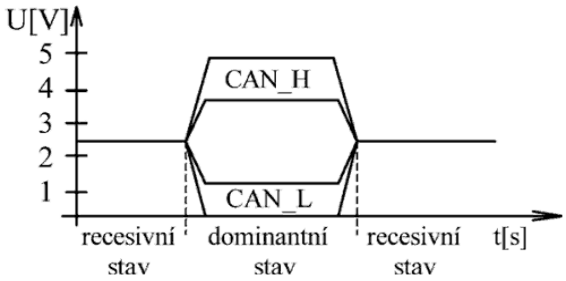
Přenosovým prostředkem je sběrnice tvořená dvouvodičovým vedením, jehož signálové vodiče jsou označeny CAN_H a CAN_L, a zakončovacími rezistory 120Ω. K této sběrnici se připojují jednotlivé komunikační uzly. Počet těchto uzlů může být až 110 (dle typu budičů CAN). Schéma zapojení fyzické vrstvy a jednotlivých uzlů je na obr. 4.3.



Obr. 4.3: Fyzická vrstva CAN [10]

Sběrnicí se přenáší dva logické stavy: aktivní (*dominant*) a pasivní (*recessive*), přičemž dominantní stav představuje logická 0, recesivní stav logická 1. Sběrnice je v dominantním stavu, je-li alespoň jeden její uzel v dominantním stavu. V recesním stavu je sběrnice tehdy, když všechny její uzly jsou v recesním stavu. V recesním stavu je rozdíl napětí mezi vodiči CAN_H a CAN_L nulový. Dominantní stav je reprezentován nenulovým rozdílem napětí. Spínače signálových vodičů jsou konstruovány tak, aby v dominantním stavu na vodiči CAN_H bylo napětí v rozsahu 3,5 až 5V, na vodiči CAN_L napětí v rozsahu 0 až 1,5V. V recesivním stavu je napětí vodičů CAN_H a CAN_L stejné a je zajištěno odporovou sítí na vstupu přijímače.

Na obr. 4.4 je na časové ose průběhu signálu znázorněno toleranční pásmo napěťových úrovní logických stavů na sběrnici CAN. Signálové vodiče CAN_H a CAN_L jsou vzájemně logicky invertované.



Obr. 4.4: Průběhy signálů CAN [10]

4.4.2 Formát rámce

Komunikační protokol CAN definuje formát přenášených zpráv na aplikační úrovni. Zprávy jsou přenášené rámcích. V definici CAN jsou určeny čtyři typy rámců:

- datový rámec (DATA FRAME)
- žádost o data (REMOTE FRAME)
- chybový rámec (ERROR FRAME)
- rámec přeplnění (OVERLOAD FRAME)

4.5 Sběrnice RS485

RS485 je standard sériové komunikace definovaný v roce 1983 sdružením EIA. Používá se především v průmyslovém prostředí. Standard RS485 je navržen tak, aby umožňoval vytvoření dvouvodičového nebo čtyřvodičového vícebodového sériového spoje. Od standardu RS232 se liší především jinou definicí napěťových úrovní, nepřítomností modemových signálů, možností vytváření sítí (též sběrnice) sestávající z až 32 zařízení a možností komunikace na vzdálenost až 1200m (proti 20m u RS232) [11].

4.5.1 Vlastnosti sběrnice

- Maximální délka sběrnice 1200 m
- Dvouvodičové nebo čtyřvodičové propojení jednotek
- Přenosová rychlost (do 10 m) až 10 Mb/s
- Až 32 připojených jednotek
- Při použití opakovačů může být počet uzlů vyšší

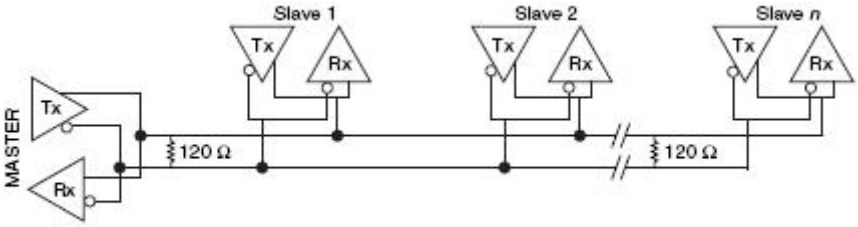
Linka používá pro přenos informací diferenciální pár vodičů s označením A a B. Při komunikaci na vyšší vzdálenosti musí být vedení na obou stranách zakončeno zakončovacími odpory, neboli terminátory. Smyslem terminátorů je zabránit odrazům signálu od konců vedení, rovněž pomáhají zvýšit odolnost linky proti rušivým signálům.

4.5.2 Fyzická vrstva

Detekce logického stavu založená na rozdílovém napětí mezi oběma vodiči je výhodná zejména kvůli eliminaci indukovaného rušivého signálu, který se většinou přičítá k oběma vodičům stejně. Přijímač rozlišuje logický stav 1 (také označovaný jako „Mark“) při rozdílu napětí A - B < -200 mV. Logický stav 0 označovaný jako „Space“ při rozdílu napětí A - B > +200 mV. Vysílač by měl na výstupu při logické 1 (klidový stav linky) generovat na vodiči A napětí -2 V, na vodiči B +2 V, při logické 0 by měl na vodiči A generovat +2 V, na vodiči B -2V.

Dvouvodičová verze RS485

Přenos je poloduplexní a proto se vyžaduje řízení přenosu dat (směru komunikace). V jednom okamžiku může vysílat nanejvýš jedno zařízení. Toto musí zajistit komunikační protokol, který však není součástí standardu RS485. Na obr. 4.5 je zapojení dvouvodičové verze RS485, kde na dvou vodičích jsou připojeny přijímače i vysílače všech jednotek. Linka je zakončena rezistory o hodnotě 120Ω.

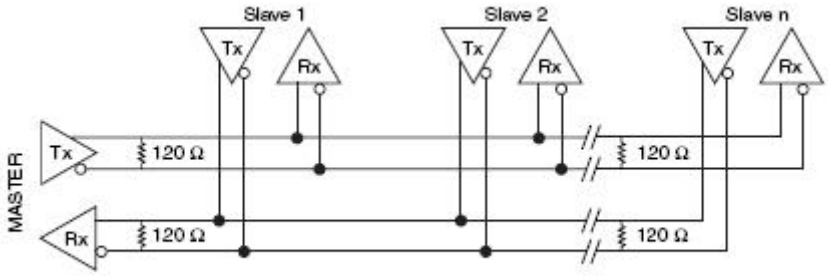


Obr. 4.5: Dvouvodičová verze RS485 [12]

Čtyřvodičová verze RS485

V některých aplikacích se používá čtyřvodičová verze RS-485, která poskytuje plněduplexní komunikaci a odpadá tak nutnost řízení směru přenosu dat. V podstatě jde o dvě dvouvodičové linky. V praxi se u čtyřvodičové linky používá i spojení 1:N, což předpokládá že slave zařízení mají schopnost odpojovat svůj vysílací kanál. Na takové lince je většinou jedno zařízení typu master, které posílá po vysílací lince příkazy a N zařízení typu slave, které přijímají příkazy a vysílají odpovědi. Výhodou je,

že master nepotřebuje přepínat směr linky a také u zařízení typu slave jsou časové požadavky na přepínání linky a na vyhodnocování příchozích zpráv mírnější. Současně nehrozí, že by slave zařízení v důsledku chyby software mohlo zablokovat „příkazový kanál“ celé sběrnice. Na obr. 4.6 je zapojení čtyřvodičové verze RS485. Každý pár je zakončen rezistory o hodnotě 120Ω.



Obr. 4.6: Čtyřvodičová verze RS485 [12]

4.5.3 Formát rámce

Samotná specifikace RS485 neříká nic o tom, jakým způsobem se mají zařízení vzájemně domluvit. Spokojíme-li se s menší rychlostí přenosu (115200bit/s, se speciálními UART obvody až 2Mb/s), můžeme s výhodou použít RS232 formát, a to jak pro bitovou tak pro bajtovou synchronizaci. Toto řešení je výhodné, obzvláště používáme-li v aplikaci platformu PC kombinovanou v kombinaci s mikrokontroléry s vlastním UART obvodem.

4.6 Výběr komunikační sběrnice

V předchozích třech kapitolách jsme si rozebrali vlastnosti uvažovaných sériových linek. Souhrn jejich vlastností je rozepsán v tabulce 4.1 pro lepší orientaci.

Vlastnost	**RS485**	**CAN**	**LIN**
Diferenciální linka	Ano	Ano	Ne
Interface	UART	CAN controller, SPI	UART
Max. Délka	1200m	1000m	40m
Max. Rychlost	10 Mbit/s	1 Mbit/s	19.2 kbit/s
Rychlost (40m)	2 Mbit/s	1 Mbit/s	19.2 kbit/s
Detekce kolize	Ne	CSMA/CD	Ne
Typický provoz	Master-slave/multimaster	Multimaster	Master-slave
Náročnost vývoje SW	Střední/vysoká	Střední/nízká	Střední/vysoká
Náklady	Střední/nízké	Střední	Nízké

Tab. 4.1: Vlastnosti sběrnic

Jak již bylo řečeno dříve, důležitými vlastnostmi jsou pro nás spolehlivost a cena. Při výběru komunikační linky ještě nebyl úplně uzavřen výběr mikroprocesoru do jednotek v kupé. Procesor MSP430, který byl zatím nejvíce ve hře ale CAN controller neobsahuje (má jen UART). Tím je CAN znevýhodněn, ale na druhou stranu, standard sběrnice řeší i přístup k fyzické vrstvě, což snižuje náročnost při vývoji software. Co se týká ceny, je na tom nejlépe sběrnice LIN a až potom RS485. Problém je v tom, že LIN není diferenciální sběrnice (menší odolnost proti rušení) a její maximální délka je 40m. Vzhledem k délce železničního vozu (25m) bychom neměli velkou rezervu, spíše žádnou. Výběr nakonec vyhrála sběrnice RS485 (čtyřvodičová) a to hned z několika důvodů:

- Rozšířená a známá průmyslová sběrnice.
- Procesor MSP430 má pouze UART
- Je odolná proti rušení (diferenciální linka).
- Má nízké náklady na realizaci (cena budičů).
- To vše je vykoupeno mírně vyšší náročností na vývoj software.

- Čtyřvodičová, protože používáme k propojení jednotek STP kabel (4 páry kroucených dvojlinek). Jednodušší na vývoj software.

4.7 Návrh komunikačního protokolu

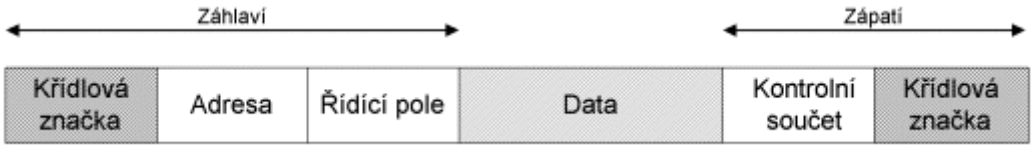
Při návrhu komunikačního protokolu byl kladen důraz na spolehlivost a zároveň jednoduchost. Z důvodu menší náročnosti na vývoj jsme chtěli pokud možno použít protokol, který nebude nijak časově závislý, to znamená, že datagramy protokolu lze v proudu dat jednoznačně rozlišit pomocí počáteční a koncové sekvence znaků. Výhodou tohoto řešení je, že mezi jednotlivými symboly v datagramu může být v podstatě libovolně dlouhá mezera a tím klesají nároky na přesné časování z pohledu mikrokontrolérů. Vhodným protokolem, u kterého jsem se inspiroval je HDLC (High-Level Data Link Control) protokol.

4.7.1 HDLC protokol

HDLC (High-Level Data Link Control) je komunikační, bitově orientovaný protokol spojové vrstvy, nadstavba protokolu SDLC, která detekuje chyby a řídí tok dat. Původně byl určen pro synchronní přenos dat, později byla norma HDLC rozšířena i pro asynchronní přenos [13].

Formát rámce

Na obr. 4.7 je znázorněn formát rámce HDLC protokolu.



Obr. 4.7: Formát rámce HDLC protokolu [14]

Křídlová značka (Flag)

Křídlová značka uvozuje datový rámec, tj. každý HDLC rámec začíná a končí právě křídlovou značkou. Na přenosové lince se mohou vyskytovat posloupnosti křídlových značek (např. v klidovém stavu). Jdou-li dvě křídlové značky po sobě, pak uvozují prázdný rámec, který se nezpracovává.

Křídlová značka se skládá z osmi bitů: 0111 1110. Šest po sobě následujících jedniček určuje právě křídlovou značku. Kdykoliv, když data na vstupu obsahují více jak pět jedniček za sebou, tak se za těchto pět jedniček automaticky vloží jedna nula. Analogicky na výstupu, je-li v přenášených datech za pěti jedničkami nula, pak se tato nula vypouští. Není-li za pěti jedničkami nula, ale jednička, pak se jedná o křídlovou značku. Tato technika se označuje jako bit stuffing [15].

Adresní pole

Adresní pole je dlouhé 8 bitů. Vyjadřuje adresu stanice, které je paket určen.

Kontrolní součet

Z přenášených dat, adresního a řídícího pole se počítá kontrolní součet, který je zpravidla buď 32-bitový nebo 16-bitový. Adresát z přijatého rámce rovněž spočte kontrolní součet, který porovná s kontrolním součtem v přijatém rámci. Jsou-li shodné, pak považuje přijatý rámec za správně přenesený. V opačném případě si u číslovaných rámců může vyžádat zopakování přenosu.

Datové pole

Datové pole obsahuje přenášená data.

Řídící pole

Podle nejnižších dvou bitů řídícího pole rozlišujeme 3 typy HDLC rámců:

- Informační rámce nebo-li I-rámce (v nejnižším bitu je 0), které jsou primárně určeny pro přenos dat. Mohou však ve svém řídícím poli přenášet i některé řídící informace (např. pozitivní potvrzení přijatých rámců).
- Nečíslované rámce nebo-li U-rámce (v nejnižších dvou bitech je 11), které se používají nejen pro přenos dat, ale i pro mnohé řídící funkce (úvodní inicializační dialog, řízení linky a diagnostiku).
- Rámce supervizoru nebo-li S-rámce (v nejnižších dvou bitech je 10). Používají se pro řízení toku dat (požadavek na vysílání, potvrzování I-rámců atd.).

S-rámce mohou být používány až když je linka inicializována, tj. když mohou být používány I-rámce. S-rámce zpravidla neobsahují datové pole.

4.7.2 Modifikace HDLC protokolu

Pro naší komunikaci budeme v procesorech používat obvody UART. Původní HDLC protokol je bitově orientovaný a mohli bychom ho použít. Problém ale je, že jeho zpracování na bitové úrovni by bylo pro použité procesory velice náročné, především pro procesor MSP430. Také implementace tohoto protokolu je poměrně složitá. Již existuje modifikovaný HDLC protokol [16], ve kterém se používá nikoliv bit stuffing, ale byte stuffing. Celý rámec začíná startovní sekvencí znaků DLE, STX (0x10, 0x02) a končí sekvencí DLE, ETX (0x10, 0x03). Pokud se někde v datech mimo začáteční a ukončovací sekvencí objeví znak DLE, musí být zdvojen, jinak by byl považován za řídící znak. Na obr. 4.8 je návrh rámce, který vznikl kombinací modifikovaného HDLC protokolu a přidáním nových (a ubráním původních) datových položek, které nejlépe vyhovovaly našim požadavkům.

DLE	STX	Typ	Adresa	Příkaz	Pořadí	Data	Součet	DLE	ETX

Obr. 4.8: Návrh komunikačního protokolu

DLE, STX (1 byte + 1 byte)
Začáteční sekvence znaků, která uvozuje celý datagram.

Typ (1 byte)
Určuje typ paketu. Pro naše potřeby používáme 2 typy paketů. Jeden je určen pro komunikaci hlavní řídící jednotky s jednotkami v kupé. Druhý typ paketu je určen pro testovací účely. Pokud máme k lince připojeno i PC, tak může hlavní jednotka posílat směrem k PC textové zprávy určené pro logování do souboru. Typ paketu pro standardní komunikaci je 0x01. Typ pro komunikaci s PC je 0x02.

Adresa (1 byte)
Adresa zařízení připojeného ke komunikační lince.

Příkaz (1 byte)
Pokud jde o vysílání směrem z hlavní jednotky do jednotek v kupé, je v tomto poli uložen příkaz. V opačném směru komunikace je toto pole inkrementováno o 1 (směr master-slave má sudé hodnoty příkazů. Směr slave-master má hodnoty liché).

Používáme 3 druhy příkazů:

- GET_STATUS (0x10) – Požadavek na zaslání aktuálních parametrů jednotky v kupé. Při odpovědi zpět je toto pole vyplněno hodnotou 0x11.
- GET_FW_VERSION (0x20) – Požadavek o zaslání aktuální verzi firmwaru. Při odpovědi zpět je toto pole vyplněno hodnotou 0x21.
- SET_PARAMETERS (0x30) – Příkaz o nastavení nových parametrů zaslaných v položce Data. V případě akceptování nových parametrů je toto pole vyplněno buď příznakem ACK (0xA1), v opačném případě příznakem NACK (0xA3).

Pořadí (1 byte)
Určuje pořadí paketu. Při každém odeslání nového paketu z hlavní jednotky je toto číslo inkrementováno. V odpovědi od jednotky z kupé je ponecháno beze změn.

Data (0 až N byte)
V tomto poli přenášíme data. Může mít délku 0 až N, kde N je pro naši komunikaci 8 byte (včetně dostatečné rezervy).

Kontrolní součet (2 byte)
Kontrolní součet přes všechny položky v datagramu vyjma začáteční a koncové sekvence datagramu.

DLE, ETX (1 byte + 1 byte)
Koncová sekvence znaků, která ukončuje celý datagram.
Hlavička datagramu má tedy velikost 4 byte a zápatí 2 byte.

4.7.3 Návrh struktury přenášených dat

Definujme si, jaká data se musí přes komunikační linku přenášet ať už z hlavní jednotky do jednotek v kupé, nebo naopak. Důležité je při návrhu myslet také na určitou rezervu pro případné přidávání dalších informací v budoucnosti.

Přenos dat z hlavní řídící jednotky do je jednotek v kupé

Z hlavní jednotky musíme umět posílat následující data:

- Stav klapky č. 1 – 1 bit
- Stav klapky č. 2 – 1 bit
- Nastavení režimu autoregulace – 1 bit
- Smazání příznaku reset – 1 bit
- Zapnutí/vypnutí LED (testovací účely) – 1 bit

Celkem tedy přenášíme 5 bitů. Datové pole datagramu tedy bude mít pouze 1 byte, přičemž rezerva zůstává 3 bity. Pokud bychom v budoucnu chtěli přenášet více informací, musíme už přidat další byte. Vše shrnuje následující tabulka 4.2.

Název/veličina	Počet bitů
Klapka 1	1
Klapka 2	1
Autoregulace	1
Reset	1
LED	1
Rezerva	3

Tab. 4.2: Formát dat - odesílání

Přenos dat z jednotky v kupé do hlavní řídící jednotky

V tomto směru přenosu je situace obdobná s tím rozdílem, že musíme být schopni přenášet informaci o teplotě v kupé a informaci o stavu otočného přepínače pro nastavení požadované teploty. Přibyly tedy datové položky:

- Teplota v kupé – 16 bitů (Čidlo teploty Dallas vrací při nastavené největší preciznosti 16 bitů ve formátu 12.4)
- Poloha přepínače – 8 bitů (Přepínač má 6 poloh, stačilo by tedy bitů méně, ale pro pohodlnější zpracování v programu přenášíme celý byte)

Zbytek položek je stejný jako v předchozím případě. Dohromady tedy přenášíme 4 znaky s rezervou 3 bity v posledním byte. Rozpis jednotlivých položek je v tabulce 4.3.

Název/veličina	**Počet bitů**
Teplota oddílu	16
Poloha přepínače	8
Klapka 1	1
Klapka 2	1
Autoregulace	1
Reset	1
LED	1
Rezerva	3

Tab. 4.3: Formát dat - příjem

Shrnutí

Používáme modifikovaný HDLC (bajtově orientovaný) protokol na fyzické vrstvě RS485 (čtyřvodičová).

Komunikace je striktně master-slave. To znamená, že komunikaci musí vždy začínat master. Slave nemá právo sám od sebe začít vysílat. Komunikace je vždy typu receive-response (na požadavek od masteru vždy existuje odpověď od slave). Protože používáme čtyřvodičovou variantu linky RS485, a komunikace je typu receive-response, nemusíme v softwaru řešit kolize na lince (master má svůj vysílací pár vodičů, na přijímacím páru vždy čeká na odpověď od dotazované jednotky).

Přenášené množství dat není velké: minimální velikost celého datagramu je 10 byte (při nulové délce datové položky a nezdvojení symbolu DLE uvnitř datagramu). Maximální velikost je 4 byte hlavička, 2 byte zápatí, 8 byte datové pole. Celkem tedy 14 byte s tím, že se všude může teoreticky vyskytnou symbol DLE, který musí být zdvojen. To je 28 byte + počáteční sekvence 2 byte + zakončovací byte = 32 byte maximálně přenesených v rámci jednoho datagramu.

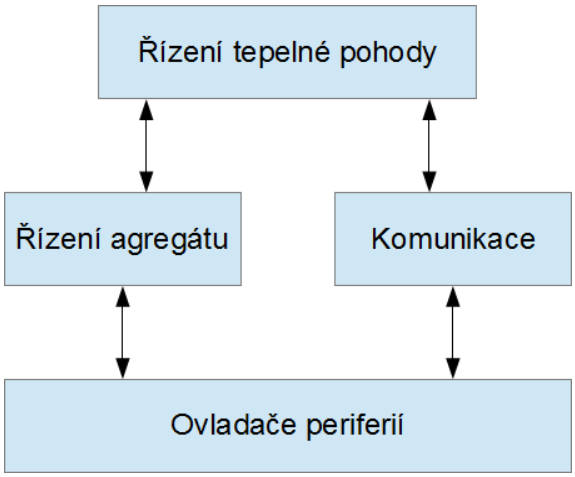
5 Návrh softwaru

V této kapitole se budeme věnovat návrhu software pro mikrokontroléry použité v tomto projektu. Software pro mikrokontroléry je napsán v objektově orientovaném jazyce C++. Nepoužíváme žádný real-time operační systém, ale program je napsán pomocí nekonečné smyčky, ve které obchází jednotlivé runtiny všech zaregistrovaných modulů.

5.1 Hlavní řídící jednotka

5.1.1 Struktura softwaru

Návrh softwaru pro hlavní jednotku je rozdělen do několika samostatných bloků tak, aby byla možnost týmové spolupráce při jejich vývoji. Na obr. 5.1 je znázorněno blokové schéma softwaru. Sestává z několika oddělených vrstev, které spolu navzájem komunikují.



Obr. 5.1: Blokové schéma softwaru pro hlavní řídicí jednotku

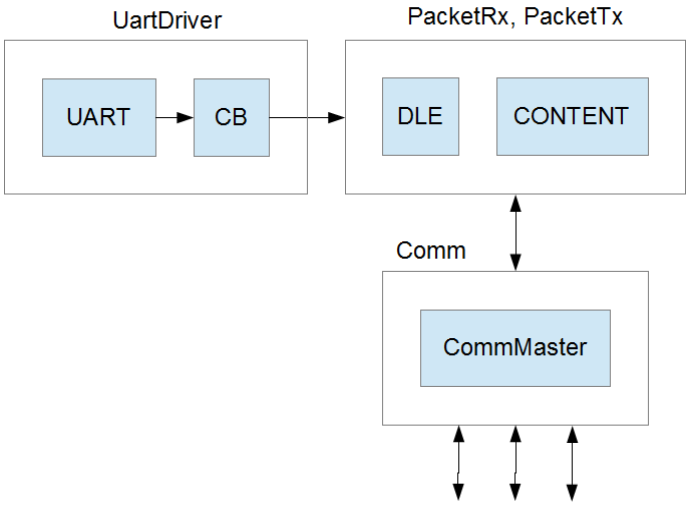
- Ovladače periferií – Objekty, které slouží k samotnému ovládání hardware. Patří zde ovladače vstupů, výstupů, AD převodníku, UART, SPI, RTC.
- Komunikace – Modul zajišťující komunikaci s jednotkami v kupé po lince RS485.

- Řízení agregátu – Tento modul se stará o chod klimatizační jednotky a topení (rozběhy kompresoru, rozběhy ventilátorů výparníku, kontrola tlaků v okruhu chladiva, spínání topných těles).
- Řízení tepelné pohody – Zastřešení všech ostatních modulů, vlastní algoritmy pro celkovou regulaci teploty vzduchu ve voze.

V našem softwarovém týmu jsem měl kompletně na starosti modul komunikace a to jak na hlavní řídící jednotce, tak na jednotkách v kupé, včetně možnosti jejich testování přímo z PC (požadavek z oddělení zkušebny ve firmě). Při vývoji na ostatních modulech jsem se podílel v rámci týmu. Samozřejmě, ne všechen software se vyvíjel od začátku. Ovladače periferií tvoří základ, který ve firmě již existuje.

5.1.2 Modul komunikace

Modul komunikace zajišťuje výměnu dat mezi hlavní jednotkou (master) a jednotkami v kupé (slave). Je navržen tak, aby kompletně zastřešoval celou komunikaci, tzn. ostatní moduly se nemusí starat o včasné a správné doručení paketů. Celá komunikace byla nejprve napsána pro platformu PC (z důvodu pohodlnějšího testování). Po důkladném otestování na PC se zdrojové kódy přeložily pro procesor DSP a MSP. Blokové schéma na obr. 5.2 znázorňuje rozdělení modulu do jednotlivých vrstev.



Obr. 5.2: Blokové schéma software pro komunikaci na straně masteru

UartDriver

Tento blok obsahuje jak ovladač hardwarového UART pro procesory MSP i DSP, tak i kruhový buffer. Do vyšších vrstev poskytuje rozhraní receiveByte(), sendByte(), receiveData(), sendData().

Při příjmu byte je v procesoru vygenerováno přerušení. V obsluze tohoto přerušení je byte vybrán z hardwarového bufferu procesoru a je vložen do kruhového bufferu pro příjem dat. Odtud si aplikace může žádat o přijatý byte.

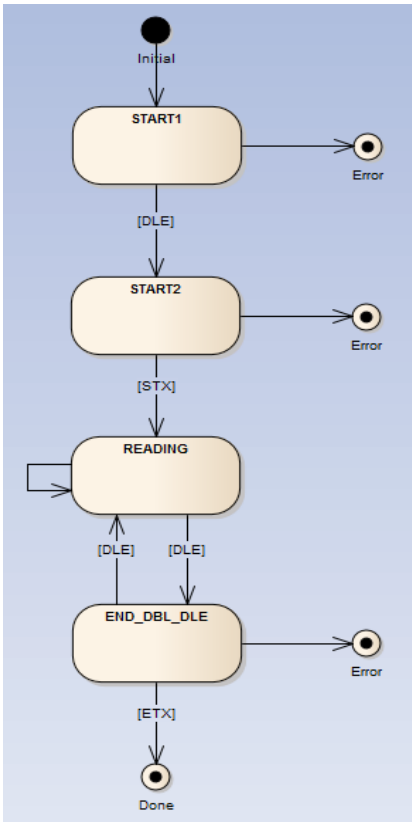
Při odesílání jsou data z aplikace vkládány do kruhového bufferu. Pokud právě neprobíhá odesílání, vyzvedne se odtud jeden byte, vloží se do hardwarového bufferu a následně se povolí přerušení od vysíláni. Při ukončení odeslání byte do sériové linky je vyvoláno přerušení. Zde se opět vyzvedne další byte z kruhového bufferu a vloží se do hardwarového. Tento cyklus se opakuje, dokud není kruhový buffer prázdný. Výhoda tohoto řešení je, že příjem i odesílání probíhá automaticky jen z přerušení procesoru. Aplikace proto nemusí nikde čekat na dokončení příjmu, ani odesíláni.

PacketRx, PacketTx

Tato vrstva zajišťuje správné parsování a kompozici paketů. Její interface je napojen na ovladač sériového rozhraní a do aplikace poskytuje metody pro aplikační vrstvu komunikace. Při příjmu jsou data nejprve zpracována přes vrstvu DLE, která obsahuje stavový automat pro parsování paketů z proudu přijímaných dat a potom přes vrstvu Content, která kontroluje správnost přijatého paketu.

Stavový automat pro příjem paketů

Na obr. 5.3 jsou jednotlivé stavy automatu pro příjem datagramů. Vysvětlení funkce automatu s jeho jednotlivými stavy je uvedeno níže.



Obr. 5.3: Stavový automat pro příjem datagramů

START1 – automat je v počátečním stavu a očekává symbol DLE (začátek paketu). Po příjmu symbolu DLE se dostává do stavu START2, jinak skončí protože nejsou splněny náležitosti pro počáteční sekvenci paketu.

START2 – pokud automat čte symbol STX, přechází do stavu READING (začátek paketu je korektní), jinak skončí.

READING – stav, ve kterém se čtou jednotlivé byte z proudu přijímaných dat a ukládají se do bufferu. Při čtení symbolu DLE přechází do stavu END_DBL_DLE. To může znamenat konec paketu, nebo jen zdvojení symbolů DLE v datech paketu.

END_DBL_DLE – při přečtení symbolu ETX automat skončí protože se jedná o konec paketu a data z bufferu předá vyšší vrstvě. Pokud je přečten symbol DLE, vrací se automat do stavu READING (jednalo se o zdvojení symbolu DLE v datech). Jakýkoliv jiný symbol znamená chybu - nejedná se o korektní paket.

Jestliže stavový automat vyhodnotí, že přijal kompletní paket, je přeposlán do vrstvy Content, která zkontroluje jeho správnost (maximální a minimální délka, kontrolní součet). Při porušení dovolené délky paketu, nebo nesouhlasu kontrolní sumy je tento paket zahozen. V opačném případě je jeho datový obsah předán do aplikační vrstvy komunikace Comm (již bez symbolů DLE, STX, ETX, kontrolní sumy a typu paketu).

Při odesílání paketů je situace jednodušší. Vrstva Content nejprve spočítá kontrolní součet, přidá ho na konec pole dat a ta předá vrstvě DLE. Tato vrstva data zabalí do paketu (přidá symboly DLE, STX na začátek a DLE, ETX na konec) a pokud se někde v datech vyskytuje symbol DLE, je zdvojen. Takto připravená data se již odesílají prostřednictvím ovladače sériové linky.

Comm

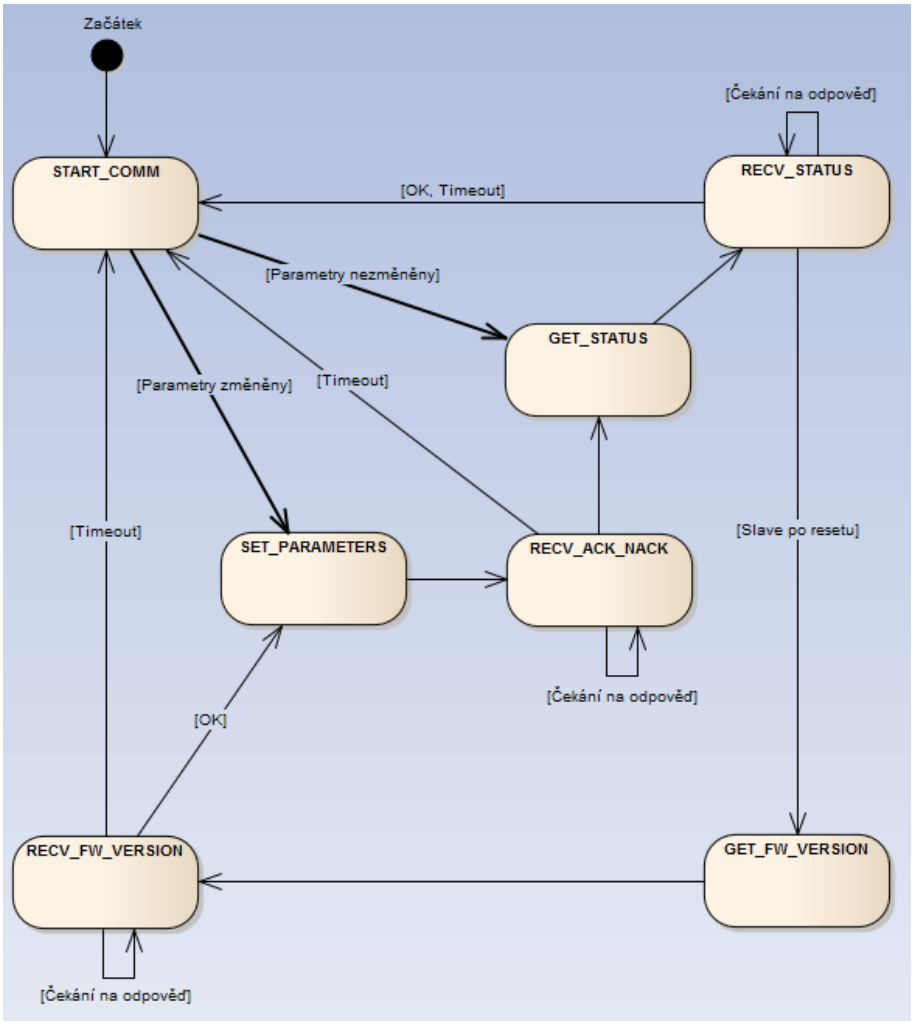
Vrstva Comm pouze zastřešuje aplikační vrstvu komunikace jak na straně masteru, tak na straně slave.

CommMaster

Aplikační vrstva komunikace na straně masteru. Princip celé komunikace s jednotkami v kupé spočívá v cyklickém dotazování se jednotek na jejich aktuální informace o teplotě, aktuální hodnotu přepínače pro požadovanou teplotu a dalšími parametry. Interakce s aplikací probíhá pomocí neblokujících metod. To znamená, že existuje mezipaměť aktuálních parametrů jednotek v kupé. Pokud se aplikace dotazuje na jakýkoli parametr, je přečten z této mezipaměti. Výhoda je v tom, že aplikace nemusí čekat až na vyřízení žádosti s pohledu přenosu dat po komunikační lince. Nevýhodou je, že právě čtený parametr nemusí obsahovat již skutečnou hodnotu. Pokud by tomu tak bylo, je hodnota parametru stará maximálně několik desítek milisekund. Tento nedostatek ale není z pohledu řízení klimatizace kritický, protože teplota v kupé se mění pomalu. Základem této vrstvy je stavový automat, který obsluhuje komunikaci se všemi jednotkami v kupé.

Stavový automat master

Na obr. 5.4 je stavový automat na straně masteru. Automat cyklicky obchází jednotlivé jednotky a dotazuje se jich na informace z kupé, popřípadě jim nastavuje aktuální parametry (klapka, LED, autoregulace). Obsahuje několik stavů, mezi kterými přechází na základě splněných nebo nesplněných podmínek. Detailnější popis jednotlivých stavů a vysvětlení jejich funkce je pod obrázkem.



Obr. 5.4: Stavový automat master

START_COMM – počáteční stav automatu. V tomto stavu se kontroluje, jestli přišel z aplikace požadavek na nastavení parametrů pro jednotku v kupé (změna stavu klapky, změna stavu LED pro ladící účely.). Pokud tento požadavek existuje, přechází automat do stavu SET_PARAMETERS, kde posílá data s novými parametry. Pokud požadavek

neexistuje, přechází do stavu GET_STATUS, kdy posílá žádost o aktuální informace z kupé.

SET_PARAMETERS – v tomto stavu odešle master dané jednotce paket s aktuálními parametry a přejde do stavu RECV_ACK_NACK, kde čeká na odpověď.

RECV_ACK_NACK - jednotka může a nemusí předchozímu požadavku o nastavení nových parametrů vyhovět. To znamená že vrací masteru paket s příznakem ACK, resp. NACK. V případě odpovědi ACK i NACK přechází automat do stavu GET_STATUS, aby se dodržela frekvence dotazování na aktuální informace z kupé. Pokud byla odpověď NACK (jednotka nechce vyhovět požadavku o nastavení nových parametrů), je tento pokus opakován znovu při dalším pokusu o kontaktování jednotky. Jestliže master nepřijme odpověď ve stanoveném časovém úseku, přechází automat do stavu START_COMM a začíná proces s následující jednotkou.

GET_STATUS – automat odešle dotaz na aktuální informace právě zpracovávané jednotky a přechází do stavu RECV_STATUS, kde čeká na odpověď.

RECV_STATUS – v tomto stavu čeká automat na odpověď k dotazu o aktuálních parametrech z kupé. V těchto parametrech je také zahrnuta informace o tom, zda-li byla jednotka v resetu (po výpadku napájení, výměna při servisu). Pokud je jednotka po resetu, přechází automat do stavu GET_FW_VERSION, jinak přechází do stavu START_COMM a začíná proces s následující jednotkou (i při timeoutu).

GET_FW_VERSION – stav pro odeslání požadavku o verzi firmwaru jednotky v kupé. Po odeslání přechází automat do stavu RECV_FW_VERSION

RECV_FW_VERSION – čekání na odpověď obsahující verzi firmwaru. Jestliže byla zpráva korektně přijata, přechází automat do stavu SET_PARAMETERS, aby mohly být jednotce nastaveny nové parametry (jednotka byla před tím v resetu). Při timeoutu nebo nekorektním paketu se přechází do počátečního stavu START_COMM a cyklus se opakuje s následující jednotkou.

Interface třídy cCommMaster

CommMaster poskytuje do aplikace poměrně široký interface pro obsluhu jednotek v kupé a pro získání informací o průběhu komunikace.

cCommMaster(UartDriver *_driver, const commSetup* const _setupPar) Konstruktor třídy cCommMaster, ve kterém se předává ukazatel na ovladač sériové linky a ukazatel na strukturu s parametry komunikace. Parametry struktury: počet jednotek v kupé, čas do kterého musí jednotka odpovědět (jinak dojte k timeoutu), rychlost cyklického dotazování se jednotek a počet pokusů o kontaktování jednotky (při překročení tohoto počtu bez úspěchu spojení se komunikace s danou jednotkou prohlásí za ztracenou).

~cCommMaster() - destruktor třídy cCommMaster

void reset() - resetování počátečních hodnot a příprava pro start komunikace

void start() - odstartování komunikace

void loop() - metoda, která se volá z hlavní nekonečné smyčky programu. Zde pracuje celý stavový automat popsaný výše

bool isConnectionEstablished(t_uint16 address) - dotaz, jestli existuje spojení s jednotkou na dané adrese

t_uint32 getSendPackets() - vrací celkový počet odeslaných paketů

t_uint32 getReceivePackets() - vrací celkový počet přijatých paketů

t_uint32 getReceiveErrors() - vrací celkový počet chyb v komunikaci

t_uint16 getTimeouts() - vrací celkový počet timeoutů

void setServo(t_uint16 address, cComm::parameters servo, bool value) - nastaví jednotce klapku do požadované polohy

void setAutoRegulation(t_uint16 address, bool value) - nastaví jednotce mód vlastní autoregulace (regulaci si řídí jednotka sama)

void setLed(t_uint16 address, bool value) - zapne nebo vypne LED na jednotce (pro testovací účely, popřípadě signalizaci stavu jednotky pro obsluhu servisu)

t_uint16 getTemperature(t_uint16 address) - vrací teplotu v daném kupé

bool getServo(t_uint16 address, cComm::parameters servo) – vrací polohu klapky v daném kupé

t_uint16 getSwitchPosition(t_uint16 address) – vrací polohu přepínače v kupé (nastavená teplota, kterou si lidé v kupé přejí)

bool getAutoRegulation(t_uint16 address) – zjistí aktuální hodnotu autoregulace

bool getLed(t_uint16 address) – zjistí aktuální hodnotu LED

t_uint32 getFwVersion(t_uint16 address) – vrací verzi firmwaru jednotky

bool isPendingParams(t_uint16 address, cComm::parameters p) – pokud aplikace nastaví jednotce nový parametr, může se touto metodou zeptat, jestli už byl jednotce doručen a byl akceptován

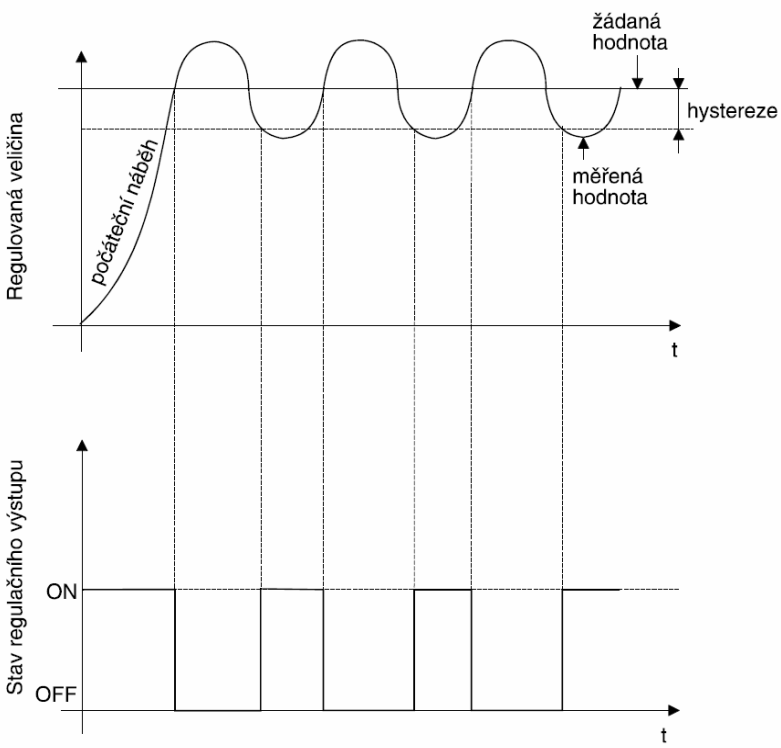
bool isPendingParams(t_uint16 address) – stejná funkce jako předchozí metoda s tím rozdílem, že se neptáme na jeden parametr, ale na všechny parametry.

bool isActualParams(t_uint16 address) – dotaz, zda-li jsou všechny parametry uložené v bufferu na straně hlavní jednotky aktuální. (je to vlastně opak metody isPendingParams() s tím rozdílem, že tato metoda bere v potaz i to, jestli jsme s jednotkou neztratili spojení)

5.1.3 Regulace teploty v kupé

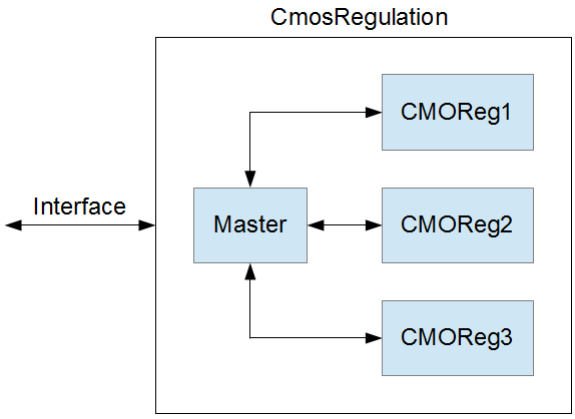
Schopnost elektroniky precizně regulovat teplotu v kupé není příliš velká. Je to z toho důvodu, že elektromagnetická klapka, která ovládá přívod vzduchu z tepelného kanálu má pouze dva stavy (otevřeno, zavřeno). Kvůli prodloužení její životnosti také nemůžeme klapku příliš často otevírat a zavírat. Z těchto důvodů jsem se rozhodl použít pro řízení teploty pouze jednoduchou, dvoustavovou regulaci.

Dvoustavová regulace se využívá pro méně náročné aplikace. Z principu není možné dosáhnout nenulové regulační odchylky. Měřená hodnota charakteristickým způsobem kmitá kolem žádané hodnoty (obr. 5.5). Regulační odchylku lze snížit zmenšením hystereze. To se však projeví častějším spínáním výkonových členů, které má nepříznivý vliv na životnost elektromechanických prvků sytému [17].



Obr. 5.5: Dvoustavová regulace [17]

Modul regulace teploty v kupé je podmnožinou modulu „řízení tepelné pohody“ z obr. 5.1. Implementace používá výše popisovaného modulu CommMaster pro komunikaci s jednotkami v kupé. Dvoustavová regulace je zde implementována tak, že master neustále vyčítá z jednotlivých kupé informace o aktuální a požadované teplotě. Tyto parametry jsou uloženy v instancích třídy cCMOReg. Každá tato instance obsahuje metodu process(), která automaticky otevírá a zavírá klapku ve svém kupé, včetně ohledu na nastavenou hysterezi a aktuální situaci (je potřeba topit, nebo chladit). Pokud není jednotka schopna „doregulovat“ teplotu na požadovanou hodnotu v určitém čase, může si aplikace vyžádat informaci, o kolik stupňů Celsia se jí tento úkol nedaří. Tato informace je důležitá pro řízení kompresoru a topných těles. Tuto implementaci zastřešuje třída cCmosRegulation. Celou situaci znázorňuje obr. 5.6.



Obr. 5.6: Blokové schéma SW pro regulaci teploty v kupé

Hysterezi, se kterou regulace počítá je možno nastavit pomocí konstanty ve struktuře *setup*. Ukazatel na tuto strukturu je předáván v konstruktoru třídy.

Interface třídy cCmosRegulation

cCmosRegulation(const setup* const _setupPar, dco::UartDriver *_driver) – konstruktor třídy cCmosRegulation. V jeho parametrech se předává ukazatel na strukturu s nastavením parametrů, především hystereze. Předává se zde také ukazatel

na UartDriver, protože objekt vytváří instanci komunikace master, aby byla zajištěna výměna informací s jednotkami v kupé.

~cCmosRegulation() - destruktor třídy cCmosRegulatio

void on() - zapnutí regulace jednote

void off() - vypnutí regulace jednote

void on(t_uint16 address) - zapnutí regulace jednotky na specifické adres

void off(t_uint16 address) - vypnutí regulace jednotky na specifické adrese

void process() - metoda volaná z nekonečné smyčky programu.

void setNominalTemperature(t_uint16 temperature) - nastavení nominální teploty v kupé. Od této hodnoty si cestující mohou nastavovat teplotu v rozmezí ± 2°C.

bool canSatisfyTemperature(t_uint16 address, t_uint16 *overflowTemperature) – metoda, která vrací, zda-li je jednotka schopna regulovat na požadovanou teplotu. Tato informace putuje do modulu řízení agregátu

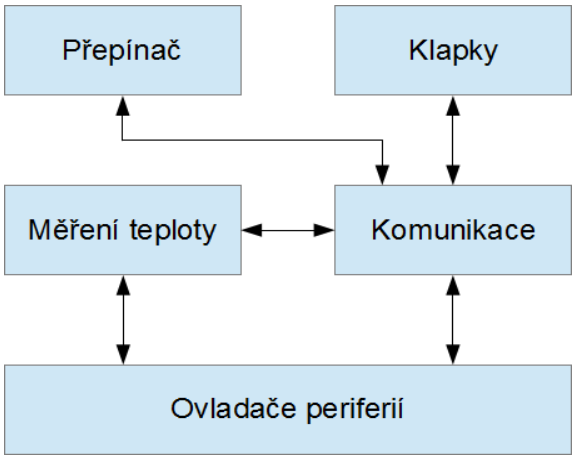
t_int16 getStatus(t_uint16 address, t_uint16 *temperature, t_uint16 *switchPosition) - Informace o aktuální a požadované teplotě v kupé.

5.2 Jednotka v kupé

Co se týká vývoje software pro jednotku v kupé, měl jsem na starosti kompletní modul komunikace a spolupracoval jsem na portování ovladače UART z procesoru DSP na procesor MSP. Zbytek modulů implementoval kolega z našeho týmu.

5.2.1 Struktura softwaru

Na obr. 5.7 je blokové schéma software pro jednotku v kupé. Sestává z modulů ovladače periferií, komunikace, měření teploty, přepínač, klapky.

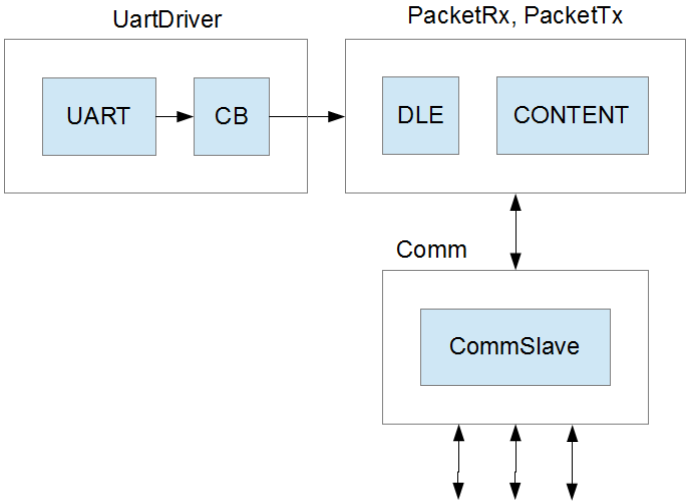


Obr. 5.7: Blokové schéma software pro jednotku v kupé

- Přepínač – modul zajišťující čtení polohy otočného přepínače pro nastavování požadované teploty v kupé. Do tohoto modulu patří i čtení fyzické adresy jednotky pro komunikaci (nastavitelná pomocí jumperů na desce)
- Ovladače periferií – zde patří především ovladač obvodu UART pro komunikaci a ovladač vstupně výstupních pinů
- Komunikace – modul pro komunikaci po lince RS485
- Měření teploty – čtení teploty s digitálního teploměru Dallas, který je připojený k procesoru pomocí sběrnice 1-Wire
- Klapky – ovládání otevírání a zavírání klapky tepelného vzduchového kanálu ve voze

5.2.2 Modul komunikace

Na obr. 5.8 je blokové schéma softwaru pro komunikaci s hlavní řídící jednotkou. Vrstvy UartDriver, PacketRx, PacketTx jsou v podstatě shodné s těmi, které jsou použity na hlavní jednotce. Pouze aplikační vrstva komunikace masteru je nahrazena vrstvou pro slave.



Obr. 5.8: Blokové schéma software pro komunikaci na straně slave

UartDriver

Funkce tohoto bloku je téměř shodná jako u ovladače UART pro hlavní řídící jednotku. Bylo ale potřeba správně nahradit hardwarové registry pro procesor MSP. Také je zde navíc ovládání vstupu ENABLE na budiči RS485, protože pokud zrovna neprobíhá odesílání dat, musí donutit přepnout budič do třetího stavu aby mohly vysílat jiné jednotky. Vstup ENABLE je tedy aktivní, jen pokud se právě vysílají data. Této skutečnosti je docíleno tím, že při začátku odesílání prvního byte se tento vstup aktivuje. Jeho deaktivace je složitější, protože procesor MSP vstoupí do přerušení od vyslaného byte již při překopírování dat z hardwarového bufferu do posuvného registru obvodu UART (odtud se už bit po bitu data odesílají z posuvného registru do komunikační linky). Důsledek tohoto je, že procesor vstoupí do obsluhy přerušení dříve než byl byte skutečně fyzicky odeslán. Proto je potřeba při posledním přerušení od vysílání (pokud

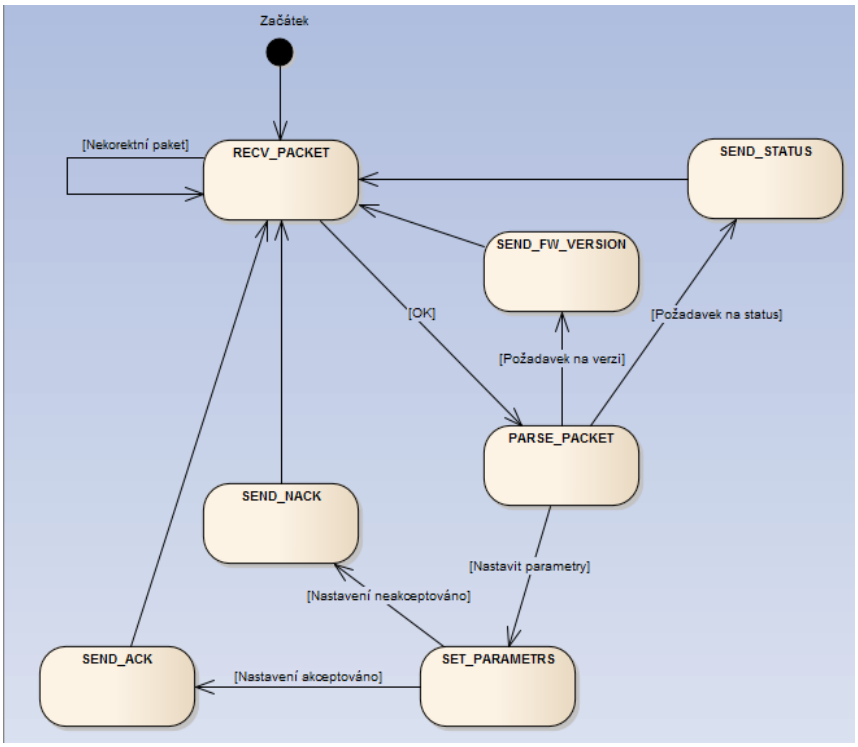
již nenásleduje další byte) nastartovat časovač procesoru přesně na dobu trvání odeslání deseti bitů (start bit, 8 datových, stop bit) při dané rychlosti komunikace. Po vypršení tohoto časovače se pak v jeho obsluze přerušení vstup ENABLE na budiči RS485 deaktivuje.

PacketRx, PacketTx

Tyto bloky jsou shodné jako u hlavní řídící jednotky.

CommSlave

Aplikační vrstva na straně jednotek v kupé. Jednotka je standardně ve stavu, kdy naslouchá lince a čeká na příchozí paket. Po přijetí paketu dochází k jeho zpracování a k přípravě odpovědi zpět do hlavní řídící jednotky. Základem této vrstvy je opět stavový automat, který zastřešuje veškerou logiku této vrstvy. Na obr. 5.9 je stavový automat aplikační vrstvy komunikace na straně jednotky v kupé.



Obr. 5.9: Stavový automat slave

RECV_PACKET – toto je počáteční stav automatu. Zde se vyčkává na příjem korektního paketu. Po příjmu takového paketu se přechází na stav PARSE_PACKET

PARSE_PACKET – první položku paketu, která se kontroluje je adresa zařízení. Pokud není paket určený dané jednotce, je zahozen a automat se vrací do stavu RECV_PACKET. V opačném případě může nastat několik situací. Hlavní jednotka požaduje nastavení nových parametrů (přechází do stavu SET_PARAMETERS), nebo se dotazuje na aktuální informace o teplotě a stavu otočného přepínače (přechází do stavu SEND_STATUS), nebo žádá verzi firmwaru v procesoru (přechází do stavu SEND_FW_VERSION)

SET_PARAMETERS – nastavení nových parametrů tak, jak byly požadovány od hlavní jednotky. Pokud jsou tyto parametry akceptovány přechází do stavu SEND_ACK. V opačném případě do stavu SEND_NACK

SEND_STATUS – jednotka sestaví paket se svými aktuálními parametry (teplota, stav přepínače, klapek, autoregulace, diagnostické LED, je-li po resetu), odešle ho hlavní řídící jednotce a automat přechází do počátečního stavu RECV_PACKET

SEND_FW_VERSION – v tomto stavu jednotka pošle paket se svojí verzí firmwaru.

SEND_ACK – nastavení nových parametrů bylo akceptováno, jednotka odpovídá příznakem ACK a vrací se do počátečního stavu RECV_PACKET

SEND_NACK - nastavení nových parametrů nebylo akceptováno, jednotka odpovídá příznakem NACK a vrací se do počátečního stavu RECV_PACKET

Interface třídy cCommSlave

Interface aplikační vrstvy komunikace na straně jednotky v kupé je podstatně jednodušší, než na straně hlavní řídící jednotky. Je to způsobeno tím, že rozhraní nemusí do programu poskytovat žádné informace o stavu měřených veličin, ani o komunikaci. Pokud přijde jakýkoliv požadavek od hlavní jednotky, samotný CommSlave informace přebírá a nastavuje v ostatních modulech pomocí jejich vlastních getterů a setterů.

cCommSlave(UartDriver *_driver, cServos *_servos, cSwitch *_posSwitch, cAutoRegulation *_regulation, cThermometer *_thermometer, cSys *_sys) Konstruktor třídy cCommSlave, ve kterém je předáván jak ukazatel na vytvořenou instanci UART driveru, tak ukazatele na jednotlivé instance modulů z obr. 5.5.

~cCommSlave() - Destruktor třídy cCommSlave

void reset() - resetování počátečních hodnot a příprava pro start komunikace

void start() - odstartování komunikace

void loop() - metoda volaná z hlavní nekonečné smyčky programu. Zajišťuje příjem paketů, veškeré zpracování požadavků a následnou kompozici odpovědi zpět do hlavní jednotky

6 Testování

Testování je jedna z nejdůležitějších částí projektu a to z jednoho hlavního důvodu. Představa, že by se dalo rychle reagovat na vzniklý problém, pokud by byl vlak vzdálený několik set kilometrů, je nereálná. Jak už bylo řečeno, měl jsem na starosti z velké části vývoj komunikace. Testování komunikace lze rozdělit do několika na sebe navazujících fází:

Testování PC – PC

Software pro komunikaci jsem začal psát dříve, než byly k dispozici z výroby první hotové jednotky. Vývoj tedy začal implementací zdrojového kódu pro platformu PC. Tato skutečnost se nedá považovat za nevýhodu, protože na PC si můžeme jednodušeji krokovat program a kontrolovat výpisy. Nejprve bylo potřeba otestovat kompozici a parsování paketů navrhovaného protokolu. Test probíhal tak, že se parser a kompozer pustili programově „proti sobě“, ještě bez účasti jakékoliv komunikační linky. Náhodně vygenerovaná data v paketu tedy musela souhlasit po průchodu těmito vrstvami. Mezi jednotlivými pakety bylo také generováno určité množství náhodných dat, která simulovala rušení na komunikační lince. Následující fází byla implementace aplikační vrstvy pro master a slave. Tady opět přistoupilo na řadu nejprve testování pouze v rámci programu, bez účasti komunikační linky. Master se v nekonečné smyčce snažil co nejrychleji kontaktovat slave a nastavoval mu nejrůznější parametry. Na ty se pak zpětně dotazoval a kontroloval, jestli souhlasí s požadovanými.

Program běžel přibližně 12 hodin a bylo přeneseno bez chyby více jak 50 miliard paketů. Potom již přišla na řadu komunikační linka RS232 a to jak pro testování pouze parseru a kompozeru datových paketů, tak pro testování vrstvy master a slave. Princip testů byl stejný jako výše zmíněný. Tímto způsobem bylo přeneseno několik desítek milionů paketů bez jediné chyby.

Testování PC – Jednotka v kupé

Portování zdrojového kódu z platformy PC na procesor MSP proběhlo bez větších potíží. Hladký průběh se ale změnil v momentě, kdy jsme na procesoru MSP spustili aplikační vrstvu pro slave a připojili ho pomocí převodníku RS232 <=> RS485 k PC, na kterém běžel master. Nejprve jsme zjistili, že jednotka v kupé sice data přijme, ale při odpovědi zpět jsou pakety špatně ukončeny (pomocí připojeného osciloskopu). Chyba byla ve špatných časových konstantách pro uvolňování linky RS485 (na budiči vstup ENABLE). Jinými slovy, jednotka se vzdala držení linky v aktivním stavu dříve, než byl odeslaný poslední byte. Spolu s tímto problémem existoval paralelně ještě další. Komunikace nám běžela pouze na maximální rychlosti 19200bit/s. To evokuje myšlenku, že jsme na hranici výkonu procesoru MSP. Po důkladné analýze jsme s pomocí osciloskopu zjistili, že obsluha přerušení pro vysílání a příjem zatěžuje procesor z více jak 50% jeho výkonu (při rychlosti 115200bit/s). Následovala tedy optimalizace kruhových bufferů, protože způsob jejich implementace nebyl optimální.

Od tohoto okamžiku se zdálo, že vše funguje spolehlivě a to i na rychlosti 115200bit/s. Celé dosavadní testování na jednotce v kupé bylo prováděno bez reálného měření teploty (jednotka odesílala smyšlenou teplotu). To znamená, že procesor nemusel obsluhovat linku 1-Wire. Po spuštění reálného měření teploty, se z výkonových důvodů nedokázal včas obsloužit buď UART, nebo 1-Wire. Po doladění obsluhy těchto periferií se podařilo jednotku dostat do stavu, kdy komunikuje na rychlosti 115200bit/s a zároveň je schopna obsluhovat všechny ostatní periferie. V této situaci jsme mohli přistoupit k rozsáhlejším testům.

Jednotka byla schopna komunikovat několik hodin bez chyby. Problém byl v tom, že vždy po čase (někdy dvě hodiny, někdy šest hodin) přestala odpovídat na příkazy od masteru. Chyba, která se takovým způsobem projevuje se hledá velmi obtížně. Po zdlouhavém hledání jsme zjistili, že pokud dojde k chybně přijatému byte (byte je znehodnocen při průchodu linkou), dostane se ovladač UART do stavu, kdy není schopen příjmu nových dat. Po odstranění této chyby a opětovném spuštění výše uvedených (úspěšných) testů, jsme mohli software na jednotce v kupé prohlásit za spolehlivý.

Testování hlavní jednotka – jednotka v kupé

Předchozí varianty testování odstranily spoustu chyb. Proto zprovoznění komunikace na hotovém hardwaru hlavní jednotky se obešlo již bez problémů. Toto testování mělo hlavně za úkol ověřit funkčnost softwaru pro master již na hotovém hardwaru, protože ve všech předchozích testech bylo zařízení master jako PC.

7 Závěr

Cílem této diplomové práce byl návrh a realizace klimatizace pro železniční vozy řady ABmz. V jednotlivých kapitolách byl popsán návrh systému od počáteční koncepce, přes definování požadavků na hardware a jeho následnou realizaci, až po návrh softwaru pro použité mikrokontroléry.

Návrh celé koncepce se ukázal jako funkční a hardware hlavní řídící jednotky splňuje všechny požadavky na řízení klimatizačního agregátu, včetně její odolnosti, která se projevila v testech elektromagnetické kompatibility.

V softwarovém návrhu se práce méně věnovala návrhu regulačním algoritmům pro řízení tepelné pohody v jednotlivých kupé. Na druhou stranu kladla větší důraz na kvalitní návrh a testování principů komunikace mezi jednotlivými celky systému. Navrhovaný komunikační protokol se během testů ukázal jako dostatečně odolný proti náhodnému rušení, které může na komunikační lince za provozu kdykoli vzniknout. Aplikační vrstva komunikace zajišťuje správnou a včasnou informovanost hlavní jednotky o aktuálním stavu veličin v jednotlivých kupé.

Během vývoje se ukázalo, že pro jednotku v kupé by bylo vhodnější vybrat výkonnější variantu mikrokontroléru. V této aplikaci jsme naráželi na výkonovou bariéru MSP430F2132 jak z hlediska velikosti paměti pro program, tak z hlediska velikosti paměti RAM. Následek této skutečnosti se odrazil v mnohem pomalejším vývoji a ladění pro tento mikrokontrolér, než jsme očekávali.

Projekt je nyní připravován k instalaci a spuštění na železničních vozech. Již teď je ale zřejmé, že práce na projektu zdaleka nekončí. Naopak, jeho dolaďování jednotlivých detailů bude probíhat ještě značnou dobu.

8 Seznam použité literatury

[1] SAJDL, Jan. *Klimatizace A/C (AC AirCondition)* [online]. 2010 [cit. 2013-05-10]. Dostupné z: http://cs.autolexicon.net/articles/klimatizace-ac-ac-aircondition/

[2] VRÁNA, Jakub: *Technická zařízení budov v praxi*; GRADA 2007; ISBN 978-80-247-1588-9

[3] SVAČINA, Jiří. *Základy elektromagnetické kompatibility* [online]. 2000 [cit. 2013-05-10]. Dostupné z: http://www.elektrorevue.cz/clanky/00041/index.html

[4] SVAČINA, Jiří. *Základy elektromagnetické kompatibility* [online]. 2000 [cit. 2013-05-10]. Dostupné z: http://www.elektrorevue.cz/clanky/00041/index.html

[5] BROK, Vladimír. *Základy elektromagnetické kompatibility* [online]. 2000 [cit. 2013-05-10]. Dostupné z: http://www.hw.cz/navrh-obvodu/z-ceho-jsou-sestavovany-prepetove-ochrany.html

[6] *LM117/LM317A/LM317-N 3-Terminal Adjustable Regulator* [online]. 2011 [cit. 2013-05-10]. Dostupné z: http://www.ti.com/lit/ds/symlink/lm117.pdf

[7] MALINA, Václav. *Poznáváme elektroniku III*. České Budějovice: Kopp, 1197.

[8] *Filter Design in Thirty Seconds* [online]. 2001 [cit. 2013-05-10]. Dostupné z: http://www.vyssotski.ch/BasicsOfInstrumentation/FilterDesignIn30Seconds.pdf

[9] SUTORÝ, Tomáš. LIN (Local Interconnect Network). [online]. 2004 [cit. 2013-05-10]. Dostupné z: http://www.elektrorevue.cz/clanky/04012/index.html

[10] TARABA, Radek. Aplikování sběrnice CAN. [online]. 2004 [cit. 2013-05-10]. Dostupné z: http://www.hw.cz/navrh-obvodu/rozhrani/aplikovani-sbernice-can.html

[11] TIŠNOVSKÝ, Pavel. Sběrnice RS-422, RS-423 a RS-485. [online]. 2008 [cit. 2013-05-10]. Dostupné z: http://www.root.cz/clanky/sbernice-rs-422-rs-423-a-rs-485/

[12] What Pins Are Needed for 2- and 4- Wire Transmission with RS-485 Serial Communication. [online]. 2008 [cit. 2013-05-10]. Dostupné z: http://digital.ni.com/public.nsf/allkb/5FFDF323C35C1B5386256F1C00622596

[13] High-Level Data Link Control. [online]. 2013 [cit. 2013-05-10]. Dostupné z: http://en.wikipedia.org/wiki/High-Level_Data_Link_Control

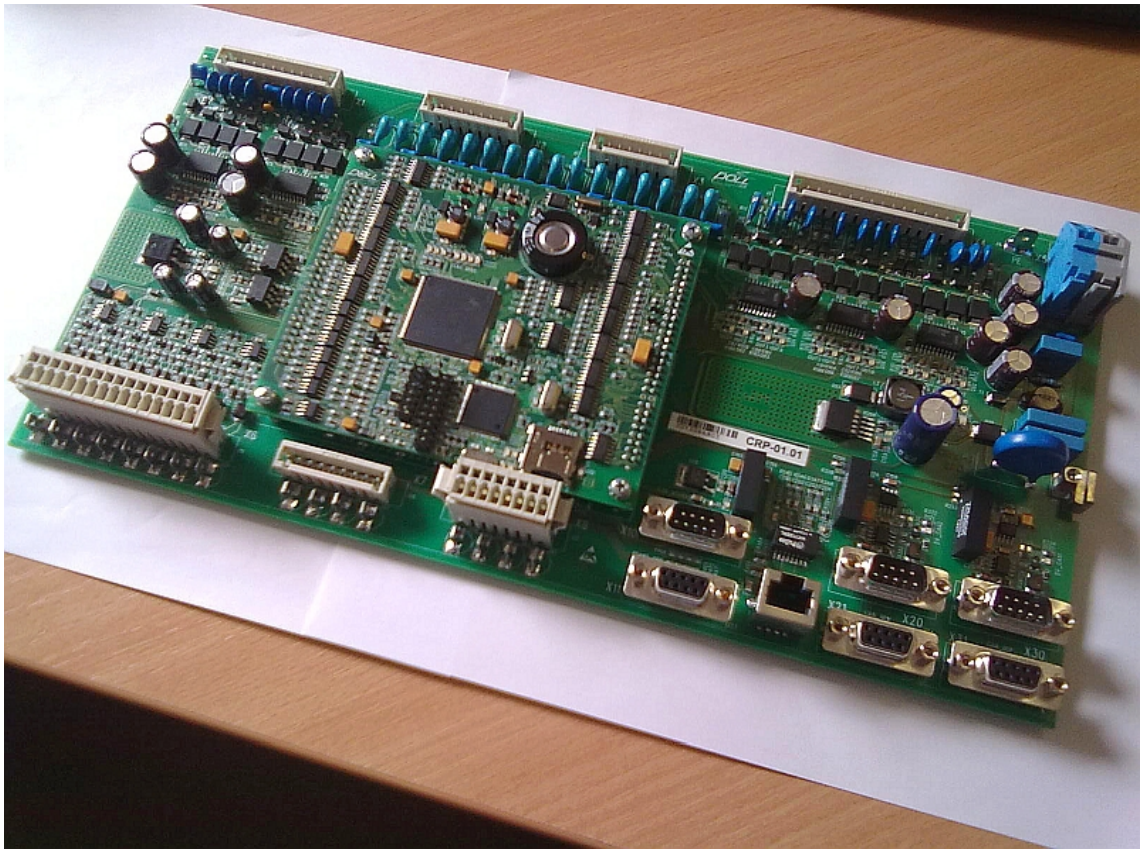
[14] Linková vrstva. [online]. 2013 [cit. 2013-05-10]. Dostupné z: http://zam.opf.slu.cz/botlik/CD-0x/4.html

[15] Bit stuffing. [online]. 2013 [cit. 2013-05-10]. Dostupné z: http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Bit_stuffing.html

[16] Bit and Byte Stuffing. [online]. 2000 [cit. 2013-05-10]. Dostupné z: http://web.cs.wpi.edu/~rek/Undergrad_Nets/C02/BitByteStuffing.pdf

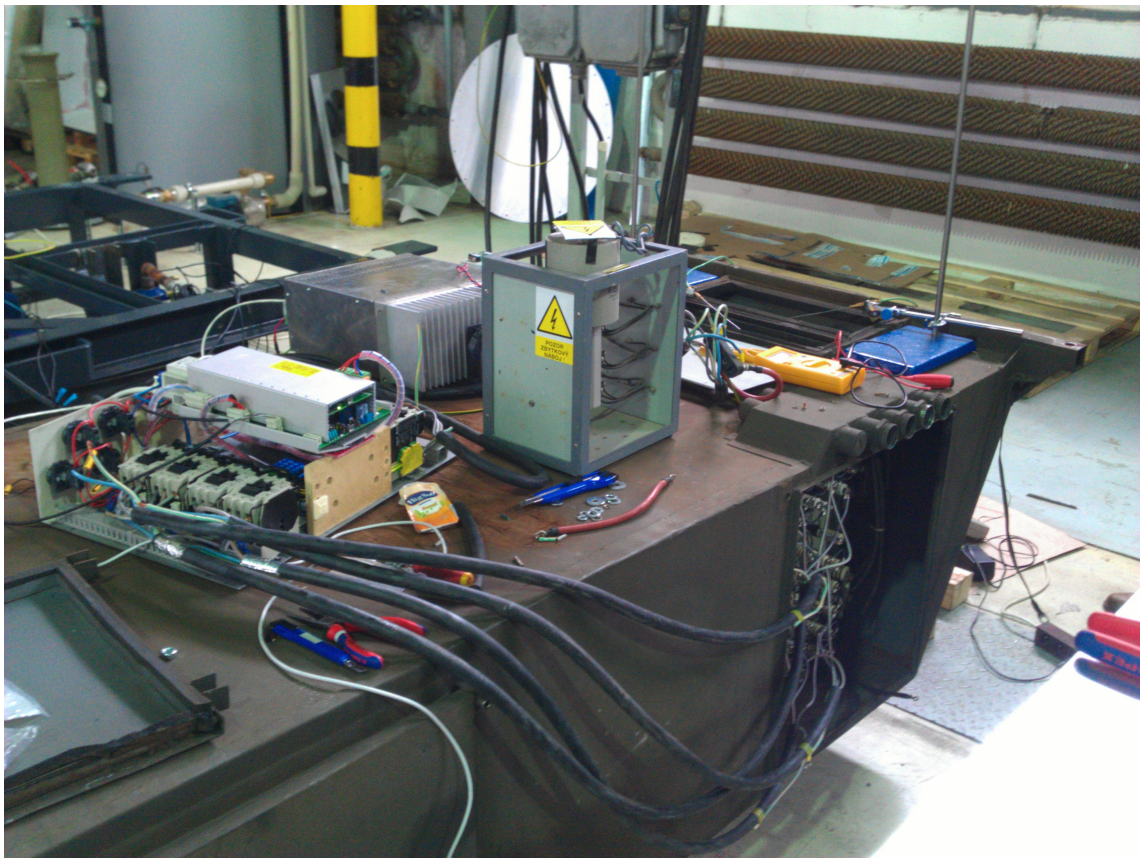
[17] Regulace. [online]. 2005 [cit. 2013-05-10]. Dostupné z: http://valter.byl.cz/sites/default/files/regulace.pdf

9 Přílohy



Obr. 9.1: Konečná podoba hlavní jednotky

Obr. 9.2: Kontejner s kompresorem



Obr. 9.3: Kontejner s výparníkem a topnými tělesy