

Zařízení s podsvětlenými tlačítky na sběrnici DALI

Device with backlit buttons for DALI bus

Bc. Jan Choutka

Diplomová práce

Vedoucí práce: doc. Ing. Zdeněk Slanina, Ph.D.

Ostrava, 2022

Zadání diplomové práce

Student:

Bc. Jan Choutka

Studijní program:

N0714A150001 Řídicí a informační systémy

Téma:

Zařízení s podsvětlenými tlačítky na sběrnici DALI
Device with Backlit Buttons on the DALI Bus

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem diplomové práce je vytvoření elektronického zařízení s až 16 podsvětlenými tlačítky pracujícím se sběrnici DALI a komunikujícím s řídicím PLC a světly s tímto standardem. Dále je potřeba zdokumentovat technologické a systémové řešení a navrhnout a sestavit software pro řízení technologie.

V souhrnu je práce charakterizována následujícími body:

1. Analýza současného stavu.
2. Obvodové a technologické řešení zařízení.
3. Realizace prototypového řešení.
4. Analýza, návrh a implementace řídicího software na vybrané hardwarové platformě.
6. Testy řešení s PLC a DALI osvětlením.
7. Zhodnocení výsledků řešení.

Seznam doporučené odborné literatury:

- [1] NORRIS Donald. *Programming with STM32: Getting Started with the Nucleo Board and C/C++*. McGraw-Hill Education TAB, 2018. ISBN 978-1260031317.
- [2] GAY Warren. *Beginning STM32: Developing with FreeRTOS, libopenm3 and GCC*. Apress, 2018. ISBN 978-1484236239.
- [3] ČSN EN 62386-101 ed. 2. Digitální adresovatelné rozhraní pro osvětlení - Část 101: Obecné požadavky - Komponenty systému.
- [4] TOOLEY Mike. *Electronic Circuits: Fundamentals and Applications*. Routledge, 2019. ISBN 978-0367421991.
- [5] KERNIGHAN Brian W., RITCHI Dennis M. *C Programming Language, 2nd Edition*. Independently published, 2021. ISBN 979-8477772476.

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

Vedoucí diplomové práce: **Ing. Zdeněk Slanina, Ph.D.**

Datum zadání: 01.09.2021

Datum odevzdání: 30.04.2022

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

prof. Ing. Jan Platoš, Ph.D.
děkan fakulty

Abstrakt

Diplomová práce se zabývá použitím sběrnice DALI a tvorbou kombinace vstupního a výstupního zařízení pro implementaci do již existujícího řešení osvětlení industriální zóny, používající sběrnici DALI. Součástí diplomové práce je návrh desky plošného spoje, softwaru ovládající vstupně výstupní periferie vytvořené DPS a ovladač (driver) pro rozhraní DALI s možností komunikace po 24-bit i 16-bit rámcích.

Klíčová slova

DALI; mikrokontrolér; osvětlení; inteligentní budovy; tlačítka; podsvětlená tlačítka; aplikační kontrolér

Abstract

The diploma thesis deals with the use of the DALI bus and the creation of a combination of input and output devices for implementation in an existing lighting solution for an industrial zone using the DALI bus. The diploma thesis includes the design of a printed circuit board, software controlling the input and output peripherals created in the PCB and a driver for the DALI interface with the possibility of communication in 24-bit and 16-bit frames.

Keywords

DALI; microcontroller; lightning; smart buildings; application controller; buttons; backlit buttons

Poděkování

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla.

Obsah

| | |
|---|-----------|
| Seznam použitých symbolů a zkratk | 8 |
| Seznam obrázků | 9 |
| Seznam tabulek | 11 |
| 1 Úvod | 12 |
| 2 Princip rozhraní DALI | 14 |
| 2.1 Napětové úrovně sběrnice DALI | 15 |
| 2.2 Typy fyzických zařízení připojitelných na DALI sběrnici | 16 |
| 3 Charakteristika současného stavu | 17 |
| 3.1 Aktuální stav provozu svítidel | 17 |
| 3.2 Navrhovaný stav | 18 |
| 4 Komunikační protokol sběrnice DALI | 19 |
| 4.1 Prevence kolize | 19 |
| 4.2 Typy zpráv | 20 |
| 4.3 Detekce kolize | 20 |
| 4.4 Obnovení po kolizi | 20 |
| 5 Výběr programovacího jazyka | 22 |
| 5.1 Jazyk C | 23 |
| 5.2 Jazyk C++ | 23 |
| 5.3 Výběr jazyka | 24 |
| 6 Návrh a výroba desky plošného spoje | 26 |
| 6.1 Výběr mikrokontroléru | 26 |
| 6.2 Zapojení tlačítek a diod do matice | 27 |
| 6.3 Elektromagnetická kompatibilita EMC | 28 |

| | | |
|----------|--|-----------|
| 6.4 | Finální zapojení DPS k podsvíceným tlačítkům | 29 |
| 7 | Softwarové řešení diplomové práce | 33 |
| 7.1 | Přehled možných způsobů komunikace zařízení | 34 |
| 7.2 | Využití přerušení pro řízení chodu programu | 36 |
| 7.3 | komunikace po sběrnici DALI | 41 |
| 8 | Ověření výsledků | 48 |
| 8.1 | Testovaná zapojení | 48 |
| 8.2 | Vyhodnocení naměřených hodnot | 48 |
| 9 | Závěr | 51 |
| | Literatura | 53 |
| | Přílohy | 55 |

Seznam použitých zkratek a symbolů

| | |
|------|--|
| BMS | – Building Management System (systém zprávy budovy) |
| DALI | – Digital Addressable Lighting Interface (digitálně adresovatelný osvět- lavací rozhraní) |
| DI | – Digital Input (digitální vstup) |
| DPS | – Deska Plošného Spoje |
| HMI | – Human Machine Interface (rozhraní pro komunikaci s uživatelem) |
| PCB | – Printed Circuit Board |
| PLC | – Programmable Logic Controller(programovatelný logický automate) |

Seznam obrázků

| | | |
|------|--|----|
| 2.1 | Napěťová úroveň na sběrnici DALI | 15 |
| 2.2 | Ukázka kódování Manchester | 15 |
| 3.1 | Stávající stav osvětlení | 17 |
| 3.2 | Požadovaný stav osvětlení | 18 |
| 4.1 | Stavový diagram odesílání zprávy | 21 |
| 5.1 | Úroveň abstrakce programovacích jazyků | 22 |
| 5.2 | Příklad spojení několika tříd návrhovým vzorem Pozorovatel | 25 |
| 6.1 | Ghosting při zapojení tlačítek do matice | 28 |
| 6.2 | Zapojení tlačítka k DPS | 30 |
| 6.3 | DALI modul V1 | 30 |
| 6.4 | Napětí na prvním modulu CH1 - DALI CH2 - Tx | 31 |
| 6.5 | DALI modul V2 | 31 |
| 6.6 | Napětí na druhém modulu CH1 - DALI CH2 - Tx | 32 |
| 7.1 | Stavový diagram sériového zpracování úloh | 33 |
| 7.2 | Use case diagram navrhovaného zařízení | 35 |
| 7.3 | Ukázka vykreslení zprávy programem třetí strany (arduinoIDE-Sériový monitor) | 36 |
| 7.4 | Problém dilatace při použití soft timer | 37 |
| 7.5 | Řešení programu kombinací pollingu a reakcí na přerušení | 38 |
| 7.6 | Diagram tříd DALI knihovny | 40 |
| 7.7 | Metoda čtení signálu manchester časovačem | 41 |
| 7.8 | Metoda čtení manchester kombinací časovače a přerušení z pinu Rx | 42 |
| 7.9 | Graf rozklíčování signálu | 42 |
| 7.10 | FSM Diagram rozklíčování signálu | 44 |
| 7.11 | Rozklíčování signálu FSM časový průběh | 44 |
| 7.12 | Chyby při odesílání dat na sběrnici | 45 |

| | | |
|------|--|----|
| 7.13 | Typy chyb na sběrnici při odesílání | 46 |
| 7.14 | Stavový diagram pro odesílání zprávy | 46 |
| 7.15 | Stavové automaty pro odesílání a kontrolu zprávy upravené pro implementaci | 47 |
| 8.1 | Využitá zapojení pro měření | 49 |

Seznam tabulek

| | | |
|-----|---|----|
| 7.1 | Značení možných přijatých stavů bitu | 43 |
| 8.1 | Zpoždění bez komunikace ostatních zařízení | 49 |
| 8.2 | Zpoždění s vlivem komunikace ostatních zařízení | 50 |

Kapitola 1

Úvod

Zadáním této diplomové práce je návrh zařízení s podsvětlenými tlačítky pro ovládání soustavy světel na sběrnici DALI. Celý návrh je řešen s ohledem na potřeby průmyslového prostředí. V diplomové práci je proveden návrh zařízení vybaveného šestnácti podsvětlenými tlačítky ošetřenými galvanickým oddělením provedeným optočleny. Zařízením může být dle přijatých zpráv po sběrnici DALI ovládáno podsvícení připojených tlačítek. Funkce podsvícení slouží jako zpětná vazba pro uživatele o stavu přiřazeného osvětlení. Řízení podsvětlení tlačítek je uzpůsobeno k ovládání centrálním uzlem, který po sběrnici DALI posílá příkazy ke změně jasu podsvětlení tlačítka.

Navržené zařízení je schopno komunikovat dvěma komunikačními protokoly. První varianta komunikace je stejná jako u zařízení typu input device dle normy 62386-103 a 62386-301. Tímto způsobem může zařízení odesílat zprávy typu event a může komunikovat s ostatními zařízeními, popřípadě být nastavováno 24-bit rámcem. Druhou variantou komunikace je komunikace odpovídající zařízení typu control gear dle normy 62386-102, který komunikuje 16-bit rámcem. Tato komunikace je poté obohacena o příkazy rezervované pro použití výrobcem zařízení. Použité příkazy slouží nejen k nastavení osvětlení, ale i k získání informací o stavu eventů z tlačítek. Tyto eventy se drží v zařízení do doby než jsou příkazem vymazány, nebo vyčteny.

V úvodní část této diplomové práce je popsán princip sběrnice DALI, jsou zde také vysvětleny možné způsoby řízení osvětlení připojeného na tuto sběrnici. Dále je zde popsána a vysvětlena možnost rozšíření a větvení sběrnice za použití uzlu, realizovaného PLC (programovatelným logickým automatem). Tímto větvením dochází k možnosti rozšíření na neomezené množství připojených zařízení.

První kapitola se věnuje popisu stávajícího stavu soustavy svítidel, v řešeném systému. Popisuje stávající soustavu řízenou centrální řídicí jednotkou kterou představuje PLC a způsob jakým stávající centrální jednotka přijímá příkazy od jednotlivých tlačítek samostatnými vodiči pomocí karty digitálních vstupů. V této kapitole jsou také shrnuty, nevýhody stávajícího systému a specifikovány požadavky na podobu budoucího systému osvětlení. Dále je tato kapitola zaměřena na řešení náhrady stávajícího centralizovaného způsobu řešení soustavy na decentralizovaný způsob komunikace

v soustavě. V této kapitole je navržený systém s tlačítky propojenými sběrnici DALI s ostatními světly a centrálním uzlem, který je zastoupeným modulem DALI pro připojení PLC.

V následující kapitole je popsáno provedení návrhu softwaru, jeho rozdělení na funkční bloky a varianty řešení objektového přístupu k programu. Dále je zde rozepsán konkrétní popis funkcí těchto bloků a jejich vzájemná komunikace. Součástí této kapitoly je také stručný přehled stavů ve kterých se může program nacházet. Tato kapitola se také zabývá způsobem, jak detekovat a vyhodnocovat seznam možných chyb vzniklých při komunikaci řídicí jednotky se sběrnici DALI a způsobem jakým tyto chyby řešit.

Následující kapitola se zabývá řešením hardwaru zařízení pro použití v průmyslovém prostředí. Řešení hardwaru obsahuje návrh způsobu připojení sériově vyráběných podsvětlených tlačítek, návrh desky plošného spoje s galvanicky oddělenými vstupními periferiemi pro připojení tlačítek a výstupními periferiemi pro připojení podsvětlení. Dále jsou zde návrhy možnosti způsobu řešení těchto periferií, jejich výhody a nevýhody. Součástí této kapitoly je i popis periferie pro připojení DALI sběrnice. Pro periferii DALI je zde vypsán způsob jakým byla periferie testována, jakých bylo dosaženo výsledků v jednotlivých verzích a v čem některé verze nevyhovovaly při testování ve skutečné soustavě.

V závěrečné kapitole je popsána konkrétní realizace projektu, způsob měření a vyhodnocení rozdílu reakční doby příkazu testovaném na různých typech komunikace a různě zatížené sběrnici DALI. Testování bylo realizováno na varianty využití zařízení přímého ovládání osvětlení nebo nepřímého ovládání osvětlení které zahrnovalo odesílání zprávy typu event a periodické dotazování na stav flag eventů v zařízení. Naměřené výsledky jsou pro porovnání a přehlednost rozepsány v tabulkách.

V závěru jsou zhodnoceny předpokládané přínosy a limity tohoto projektu. Jsou zde také předloženy návrhy na úpravu dosavadního systému, které dle doložených měření v této práci mohou zlepšit rychlost celé komunikace probíhající na soustavě DALI.

Kapitola 2

Princip rozhraní DALI

Rozhraní DALI (Digital Addressable Lighting Interface) je adresovatelné rozhraní pro komunikaci převážně mezi vypínači a řadiči osvětlení určené pro automatizaci osvětlení výrobních hal, stadionů, osvětlení kanceláří nebo domácího osvětlení.

Soustavy založené na systému DALI řeší problematiku složitějších osvětlovacích systémů pro implementaci individuálních sběrníkových systémů a zjednodušuje ovládání pro koncového uživatele. Rozhraní DALI vzniklo jako odezva na potřebu všestrannějšího ovládání. Systém s DALI je adekvátní náhradou přepínačů analogové signalizace, jako jsou ovládací prvky stmívání, přepínačů, vypínačů apod.

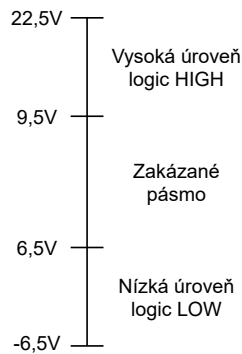
Systém DALI poskytuje kompromis mezi běžným ovládáním, pomocí vypínačů a analogových prvků stmívání (které mají omezené možnosti ovládání) a mezi systémy správy budov BMS (building management systems), jako je např. KNX, Control4, Loxone a další, které zahrnují nejen osvětlení ale, i vytápění, ventilaci, klimatizaci, ovládání žaluzií a podobně. Pro řízení pouze osvětlení jsou tyto systémy z důvodů své komplexnosti velmi nákladné a složité. V případě požadavku řízení pouze osvětlení je aplikace DALI rozumným řešením.

Jednou z hlavních výhod DALI systémů je provázanost mezi světly navzájem a připojením na centrální ovládání. Neznamená to ovšem, že by systém nemohl fungovat v malých uzavřených okruzích, nebo by byl ovlivněn výpadkem centrálního ovládacího uzlu, popřípadě poruchou na vedení způsobenou rozpojením vodičů. Je-li tato sběrnice připojena ke zdroji, může probíhat ovládání jako decentralizovaný systém.

Další výhodou, je v případě potřeby, možnost sběrnici DALI propojit s některým ze systémů BMS a rozšířit tak její funkčnost. Ani v tomto případě však není nutností využívat propojení s nadřazeným systémem a v případě odpojení od tohoto systému, celá soustava osvětlení připojena na sběrnici DALI může pracovat dále jako decentralizovaný systém [1, 2, 3].

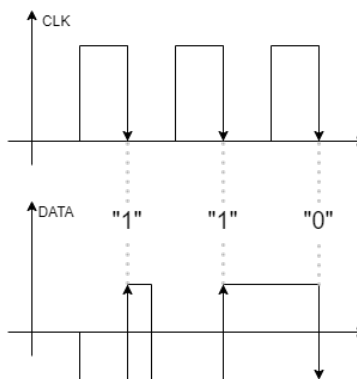
2.1 Napěťové úrovně sběrnice DALI

Sběrnice DALI používá kódování Manchester dle IEEE 802.3. Toto kódování nevyužívá pro logické úrovně vysokou nebo nízkou hodnotu napětí, ale náběžnou nebo sestupnou hranu signálu. Logické "1" je přiřazena náběžná hrana a logické "0" odpovídá sestupná hrana. Napětí musí projít při přechodu zakázaným pásmem a ustálit se na vysoké nebo nízké log. úrovni viz. obr. 2.1.



Obrázek 2.1: Napěťová úroveň na sběrnici DALI

Stav na sběrnici, kdy je napětí beze změny ve vysoké úrovni se označuje jako IDLE a tento stav odpovídá stavu sběrnice v klidu (bez komunikace). Pokud tento stav nastane na konci komunikace po dobu delší než 2400 us znamená to odeslání stop bitu, neboli posledního bitu a uvolnění sběrnice pro další komunikaci. Start bit naopak nastává při přechodu ze stavu IDLE skokem do nízké úrovně a zpět na vysokou úroveň za dobu $\frac{1}{2}$ bitu viz. obr. 2.2.



Obrázek 2.2: Ukázka kódování Manchester

Výhodou tohoto kódování je, že se může tento přechod z jedné úrovně na druhou použít k synchronizaci rychlosti čtení zpráv, jelikož je přechod mezi úrovněmi vždy uprostřed bitového intervalu. Z tohoto důvodu také nemůže dojít k chybnému načtení počtu bitů u řady stejných po sobě následujících hodnot.

2.2 Typy fyzických zařízení připojitelných na DALI sběrnici

Na sběrnici DALI může být připojeno až 63 zařízení s krátkou adresou začínající nulou a končící adresou 62 (0b111110), kdy adresa 63 (0b111111) slouží pro broadcast. Další možnost rozdělení je dle skupin pomocí skupinové adresy zařízení. V těchto skupinách lze volat a nastavovat jejich hodnoty současně. využitím skupin se značně zrychlí přenos dat a reakce na změnu v systému. Další výhodou je uvolnění zatížení sběrnice DALI. Připojená zařízení na sběrnici DALI jsou označována jako: Control gear, Input device, Application controller a Bus power supply. Veškerá zařízení podléhají normě IEC62386-101 [4].

2.2.1 Control gear

Control gear představuje zařízení, která jsou řízena sběrnici DALI. Mezi tyto zařízení se řadí např. světla popřípadě jiná osvětlovací zařízení. Pro každý typ tohoto zařízení existuje příslušná norma IEC62386-2xx, kde x označuje typ zařízení. Tato zařízení přijímají šestnácti bitové rámce a mohou odeslat odpověď ve formě pouze osmi bitového rámce.

2.2.2 Input device

Input device představuje zařízení, které předává informace po sběrnici DALI o prostředí ve kterém jsou umístěny. Takovým zařízením může být např. tlačítko, senzor osvětlení, přítomnosti senzor a další. Dle normy IEC62386-103 by měl obsahovat "multimaster transmitter timing" (časový protokol pro více zařízení) a standardně odesílat pouze 24 bitový rámeček.

2.2.3 Application controller

Aplikační kontrolér má dva druhy provozu Single-master a Multi-master. Single-master pracuje s jiným časovým protokolem než Multi-master. Aplikační kontroler může obsahovat několik instancí input device, např. tlačítka, analogové ovladače a další. Aplikační kontrolér je možno využít na rozdíl od input device ke zpětnovazebnímu řízení osvětlení. Další možností je řízení v závislosti na snímání slunečního svitu. Porovnáním s input device podporuje aplikační kontrolér navíc kromě 24-bit rámce i 16-bit rámeček a osmibitovou odpověď.

2.2.4 Bus power supply

Bus power supply udržuje napětí na sběrnici a napájí rozhraní připojených zařízení na tuto sběrnici. Může být připojeno jedno nebo více těchto zdrojů, v případě více připojených zdrojů je potřeba provést nastavení pro zajištění maximálního proudu vysílajícím rozhraním. Maximální proud nesmí přesáhnout 250 mA. viz. ČSN EN 62386-101 4.7.2

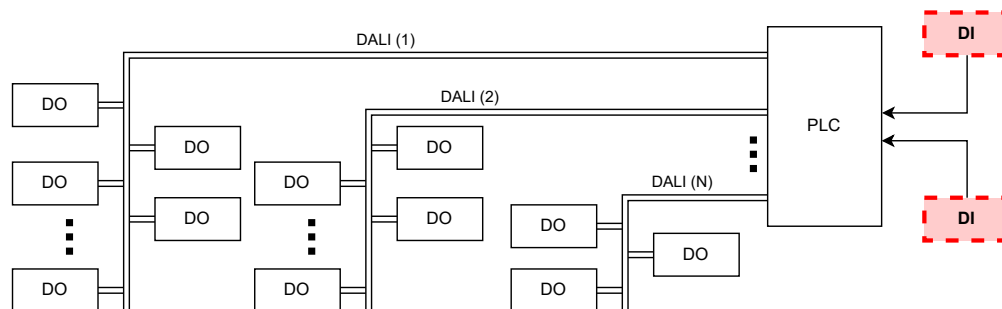
Kapitola 3

Charakteristika současného stavu

Úkolem diplomové práce je navrhnout náhradu stávajícího řešení připojení tlačítek v soustavě osvětlení existujícího systému. Nahrazením by došlo ke zjednodušení stávajícího systému i následujících projektů, snížení ceny, snížení počtu vodičů nutných k zapojení a tím i zvýšení rychlosti a zjednodušení montáže systému.

3.1 Aktuální stav provozu svítidel

V současnosti je provozováno na průmyslovém objektu osvětlení z centrálního bodu, kterým je PLC s několika rozšířeními pro připojení sběrnice DALI. Každé rozšíření vytváří samostatnou větev a větve jsou propojeny logikou řídicí jednotky PLC. Na každé této větvi je umístěno několik koncových zařízení "Control gear" (světelných) označenými DO (DALI Output) viz. 3.1 Ovládání je řešeno pomocí programu v PLC, které po sběrnici DALI posílá příkazy k rozsvícení, nebo zhasnutí světelných do každé větve samostatně. Tímto je umožněno obsloužit více než je maximální počet světelných na sběrnici DALI. Maximální počet zařízení připojitelných na sběrnici DALI je 63 viz. kapitola 2.2.



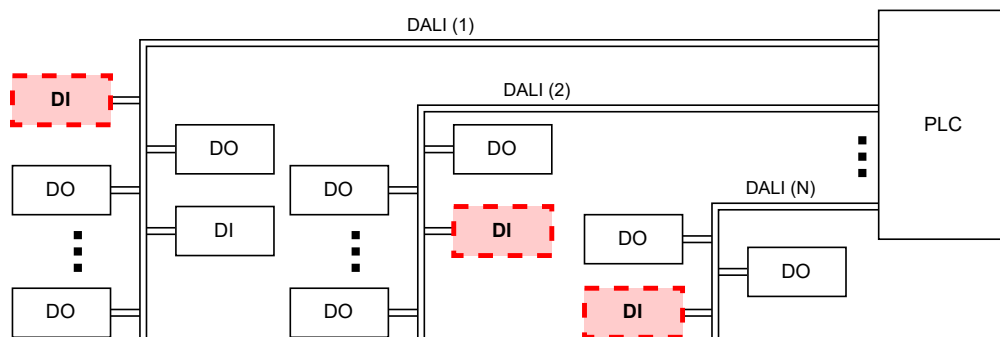
Obrázek 3.1: Stávající stav osvětlení

3.2 Navrhovaný stav

Návrh řešení obsahuje připojení zařízení s tlačítky do větví s osvětlením. Toto zařízení bude mít možnost komunikovat s osvětlením ve větví, kde je vestavěno, anebo odesláním eventu bude informovat hlavní uzel připojený DALI modulem k PLC. Hlavní uzel poté rozdistribuuje požadavek uživatele do ostatních větví. Zařízení s tlačítky bude možno kontaktovat stejným komunikačním protokolem jako osvětlení a bude mu možné po sběrnici DALI ovládat podsvícení tlačítek, čímž bude signalizovat zapnutý nebo vypnutý stav osvětlení.

Pro otestování je nutné vypracování návrhu DPS s možností připojení podsvětlených tlačítek a periférií DALI pro ovládání osvětlení připojitelného ke sběrnici DALI, která je již v objektu implementována. Z důvodu použití v průmyslovém prostředí je vhodné, aby deska plošného spoje velikostí umístitelná do krabičky zachytitelné na DIN lištu. Veškeré periferie DPS musí být z důvodu ochrany ošetřeny galvanickým oddělením. Napájení zařízení je řešeno z místního rozvodu o nominálním napětí 24 V.

Celé zařízení je v souladu se zásadami EMC. V diplomové práci je porovnání řízení osvětlení metodou přímého ovládání světel, metodou odesílání eventů a metodou periodického dotazování centrálním modulem (PLC řídící celé osvětlení).



Obrázek 3.2: Požadovaný stav osvětlení

Kapitola 4

Komunikační protokol sběrnice DALI

Původní verze DALI, označována též někdy jako DALI1, byla připravena na jediný řídicí modul (režim single-master), který ovládá veškerá osvětlovací tělesa připojena k této sběrnici. Vydáním druhé verze označené jako DALI2 došlo k rozšíření o normu ČSN EN 62386-103, a tím vznikají nové možnosti rozšíření při využití sběrnice s novými vstupními zařízeními. [5]

Pro využití nových vstupních zařízení je nutné, aby všechny potřebné zařízení propojené sběrnici DALI obsahovaly protokol pro DALI2 tzn. časové protokoly a dvaceti čtyř bitové rámce. Dalším krokem je nutnost použít nastavení vstupních zařízení v režimu multimaster. Tímto krokem se změní a upraví i některá nastavení týkající se časování komunikace. Aby nedocházelo k narušování posílání zpráv z několika různých zdrojů, řídí se všechny zařízení na sběrnici DALI následujícími pravidly popsanými v normě ČSN EN 62386-103 a ČSN EN 62386-101 [6, 7, 8]

4.1 Prevence kolize

Prevence kolize se vztahují pouze na "forward transmission" tzn. šestnácti a dvaceti čtyř bitové rámce u multimaster zařízení. Všechna zařízení připojená na sběrnici se vždy snaží vyhnout kolizi. Vyhnutí se kolizi znamená, že všechna zařízení musí zpětně naslouchat i datům, které odesílají a v případě, že zjistí neshodu v odeslaném a přijatém bitu, ukončí odesílání komunikace a vyčkávají na další příležitost k odeslání. Po poslední zaznamenané komunikaci, se zahájí opět odesílání poslední přerušené zprávy od začátku. Pokud je odesílaná zpráva poškozena, a hrozí změna významu odesílané zprávy je třeba použít obnovovací mechanismus. Tento mechanismus iniciuje zařízení, které zprávu poškodilo. Obnovovací mechanismus zpráv je založen na principu snížení napětí na sběrnici po stanovenou dobu a všechna zařízení připojená na sběrnici tím upozorní na poškozenou zprávu. Nízkým napětím po tuto stanovenou dobu není možné dokončit odesílání poškozené zprávy, čímž je tato zpráva ztracena [1, 9].

4.2 Typy zpráv

Všechny zprávy komunikace po sběrnici mezi jednotlivými zařízeními se skládají z jednoho start bitu vysvětleného v kapitole 2.1, n datových bitů v závislosti na typu zprávy a dva stop bity označující konec zprávy.

4.2.1 Odpověď

Odpověď na dotaz je nejmenší datový rámec zprávy, který obsahuje osm datových bitů a neobsahuje cílovou adresu. Odpověď - je vždy odeslána na základě přijetí instrukce typu query (požadavek) viz. 4.2.2. Datový rámec typu "Odpověď" je označován jako backward frame (zpětný rámec) a není přerušitelný jinou zprávou [4].

4.2.2 Instrukce

Instrukce a požadavky jsou zprávy obsahující 16 datových bitů. Instrukce mění stavy definovaných registrů zařízení, čímž nastavují chování zařízení, například: zapnutí, vypnutí, nastavení maximálního výkonu osvětlení, nastavení aktuální scény a další. Požadavky z těchto registrů požadují hodnoty ve formě zpětných odpovědí hodnoty registru nebo formou ano/ne - 0/255. viz. 4.2.1 [4].

4.2.3 Event

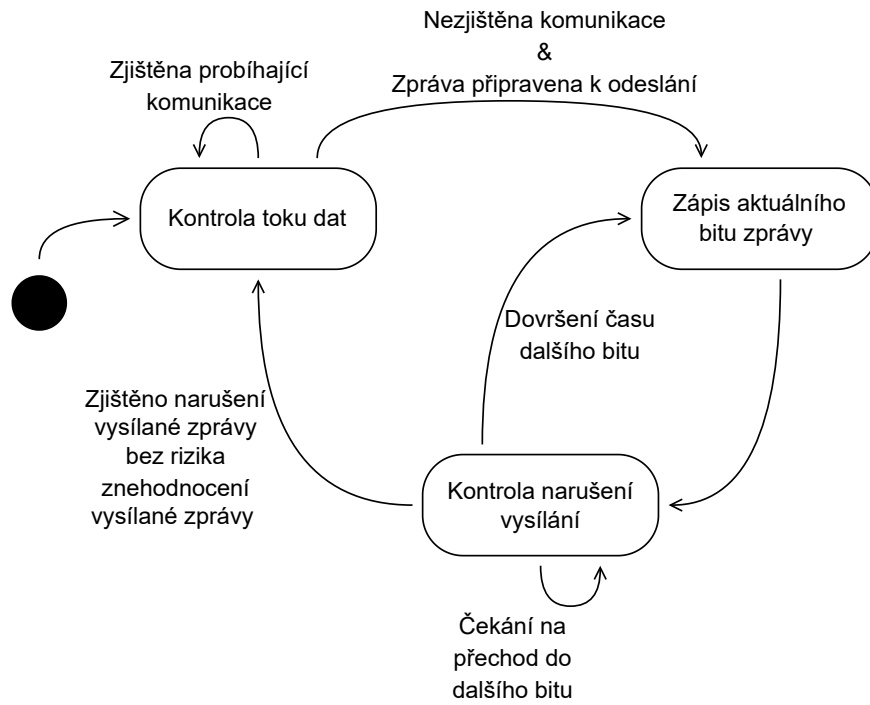
Event se používá pouze u zařízení DALI2, což jsou tlačítka a další senzory sloužící k aktivní změně osvětlení (zpráva odchází přímo z těchto zařízení bez toho, aby byly dotazovány na svůj stav). Eventy používají 24bitový rámec. Tento rámec obsahuje vlastní adresu zařízení, které zprávu odesílá, typ eventu ke kterému došlo a instanci, která jej vyvolala. [4]

4.3 Detekce kolize

Aplikuje se na forward frame, při odesílání dat sleduje hodnotu napětí na sběrnici. V případě, že není provedena změna v požadovaném časovém intervalu od zadání požadavku změny, je tento rámec zprávy okamžitě ukončen. V případě, že při kolizi existuje možnost narušení některého ze stanovených časů (destroy area) u předchozích odeslaných bitů, spustí se obnovovací funkce sběrnice. Poté může zařízení znovu odeslat tuto přerušenu zprávu. [4]

4.4 Obnovení po kolizi

K obnovení dochází po zjištění narušení zprávy a znehodnocení narušených dat. Při znehodnocování sběrnice sníží napětí na nízkou hodnotu po dobu "break time"(1,2 ms - 1,4 ms). Tato akce je se používá pouze ve verzi DALI2 a režimu multi-master bus. Po obnovovacím mechanismu dochází k



Obrázek 4.1: Stavový diagram odesílání zprávy

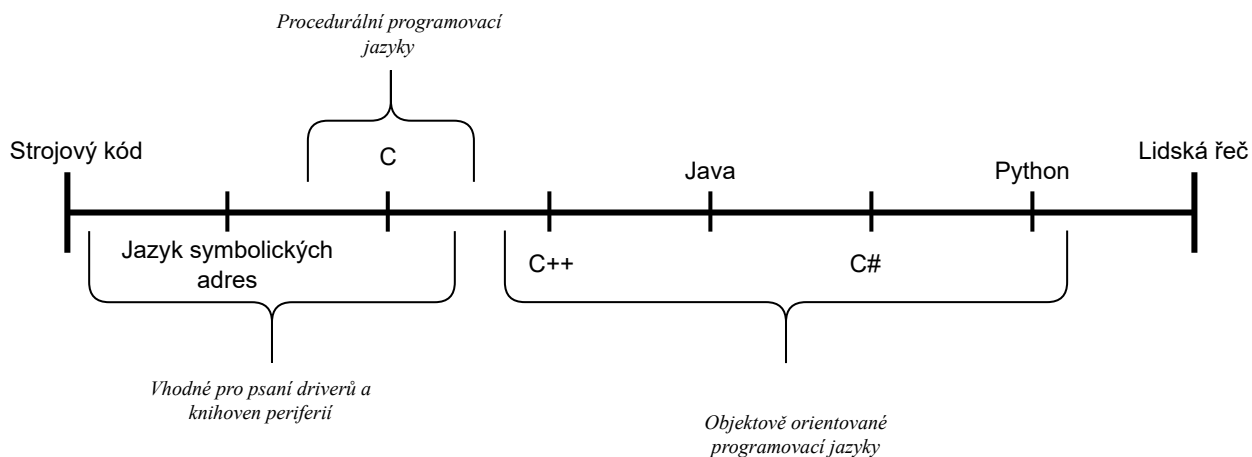
odesílání dat standardně, jako při inicializaci sběrnice a nová zpráva začne náhodně mezi (4 ms - 4,6 ms) dle priority zprávy. [4, 8]

Kapitola 5

Výběr programovacího jazyka

Programy pro mikrokontrolery se programují převážně jazyky C a C++. Tyto programy je možno používat každý zvlášť, případně je kombinovat. Každý z těchto jazyků má své klady a zápory a je proto vhodné volit programovací jazyk na základě budoucích vlastností projektu. Platforma STM32 Cube IDE zvolená pro tento projekt podporuje oba tyto jazyky a z tohoto hlediska nejsme omezeni vývojovým prostředím. [10]

Programovací jazyky se rozlišují dle mnoha parametrů, kdy jeden z nejdůležitějších parametrů je úroveň abstrakce. Úroveň abstrakce vyjadřuje, zda se jazyk spíše blíží strojovému kódu nebo mluvené řeči. Jazyk s nejnižší úrovní abstrakce je strojový kód, a naopak s nejvyšší úrovní abstrakce je lidská řeč. Na obrázku viz. obr. 5.1 vidíme přibližný přehled několika základních jazyků. Tento přehled nám dává představu, v jakých programovacích jazycích bychom měli tvořit program. [11]



Obrázek 5.1: Úroveň abstrakce programovacích jazyků

5.1 Jazyk C

Jazyk C je procedurální programovací jazyk. Program je rozdělen do procedur (funkcí), které se píše zvlášť a spouští se z hlavního programu, nebo z jiné procedury. Jazyk C je kompilovaný jazyk, což znamená, že není možno jej spouštět jako zdrojový kód, ale musí projít kompilací. Proces převodu na strojový kód probíhá v procesech preprocesoru, kompilátoru, assembleru a linkeru, čímž je vytvořen výsledný strojový kód pro konkrétní mikrokontrolér, který je možno do tohoto mikrokontroléru nahrát. Program je touto úpravou podstatně menší a rychlejší, než kdyby mikrokontrolér musel zpracovávat kód z textu, jak je tomu u interpretovaných jazyků. [12]

5.2 Jazyk C++

Jazyk C++ je z pohledu abstrakce výše postavený jazyk než jazyk C a je objektivě orientovaný. Znamená to, že jeho hlavními bloky není procedura, ale Třída (objekt). Třídy v sobě sdružují proměnné používané pro instanci tohoto bloku a metody (funkce uvnitř třídy, které mohou pracovat se všemi jejími parametry). Tento jazyk je vhodný zejména pro obsáhlejší projekty a zároveň pro práci ve větším kolektivu, kdy každý programátor může vytvářet a pracovat na své třídě. Struktura objektivě orientovaných programovacích jazyků musí splňovat podmínky:

5.2.1 Zapouzdřenost

Proměnné uvnitř třídy nejsou viditelné pro ostatní třídy, k těmto hodnotám je možné přistupovat pouze pomocí metod, které třída poskytuje. Metody samotné musí být ošetřeny specifikací, pro které typy tříd je viditelná a které třídy ji mohou volat. Využitím zapouzdření dochází k vytvoření kódu který je znovupoužitelný v dalších projektech. Takto vytvořený blok je možno vzít a přesunout bez problémů do jiného projektu, protože má definovaný interface, kterým přenáší data do a z třídy.

5.2.2 Rozšiřitelnost

Rozšiřitelnost (abstrakce) je důležitou vlastností pro udržení přehlednosti programu po delší dobu s možností provádění změn bez toho, aby se program stal nepřehledným a nadále neupravitelným. Rozšiřitelnost udává, že by veškeré složitější funkcionality uvnitř třídy měly být děleny na další třídy tak, aby ve výsledném kódu, každá třída měla pouze jednu funkci. Pro další funkcionality by třída měla využívat další třídy. Tímto se vytvoří přehledné stromové uskupení, kde musí být velmi jednoduché přepracovat jeden objekt, jelikož již není zatížený dalšími funkcionalitami, které není potřeba změnit.

5.2.3 Dědění

Třída, která je již jednou vytvořena, může být použita jako vzor pro vytvoření dalších tříd, které děděním využívají funkce již vytvořené třídy, aniž by bylo nutno je kopírovat do nové třídy. Takovéto třídy pak označujeme jako Potomek. Rodič je třída poskytující metody a Potomci tyto metody využívají popřípadě je pro sebe mohou měnit.

5.3 Výběr jazyka

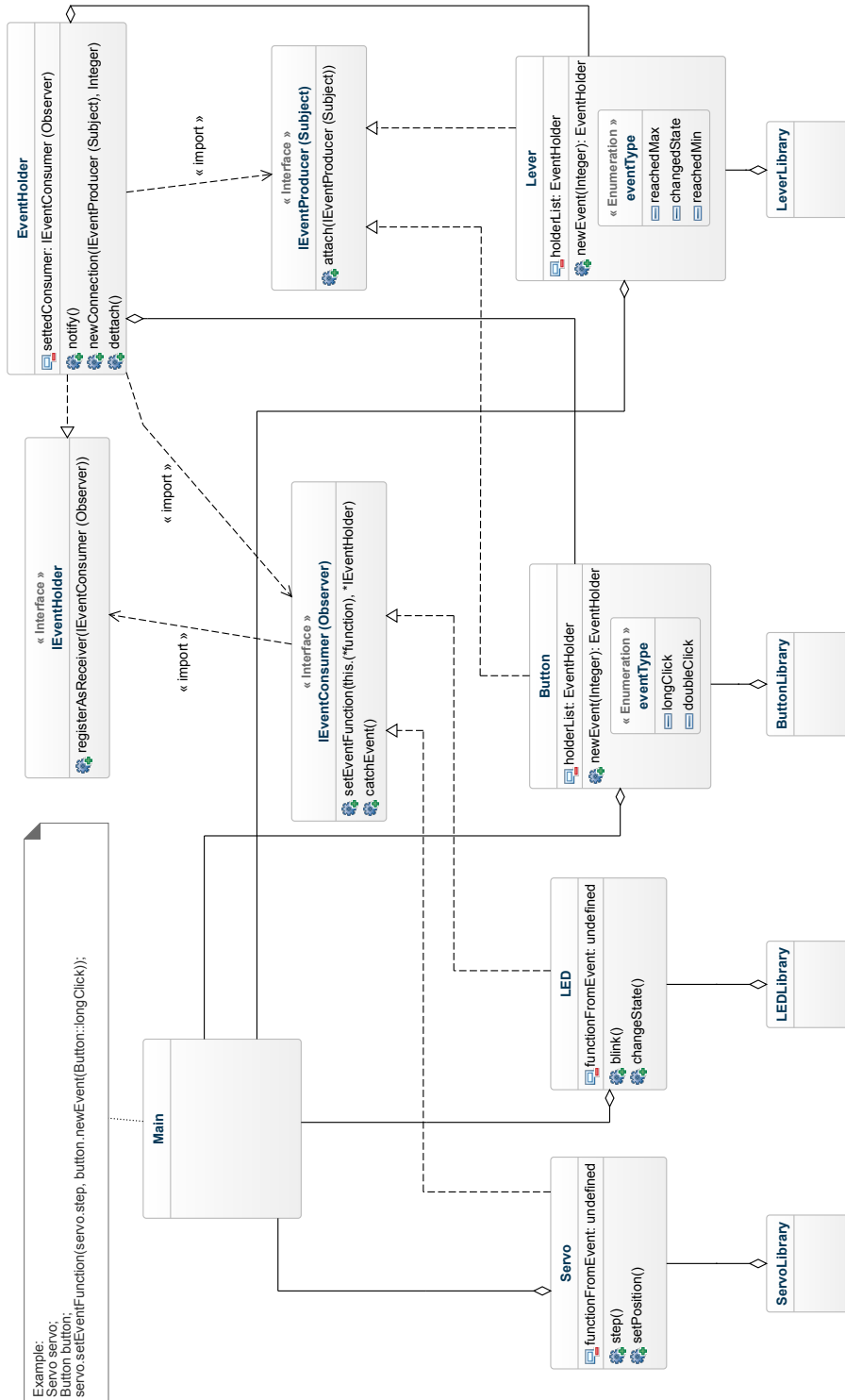
Vzhledem k rozlehlosti programu tvořeného pro zadaný cíl této diplomové práce se jeví jako vhodné použití jazyka C++. Tento software by byl rozdělen do bloků na třídy stejně, jako vytvořený HW. Při vytváření bloků a propojení mezi nimi na stejné úrovni se začne projevovat velký nárůst kódu. Je velice komplikované jednoduše vytvořit dvě třídy na stejné úrovni, které komunikují oběma směry. Jedním z pokusů řešení bylo dle návrhového vzoru Pozorovatel. Tento návrh zahrnuje sadu tříd vzájemně komunikujících skrze interface a posílající eventy viz. diagram 5.2. Takto sestrojený program by v budoucnu vytvářel zbytečně velké množství kódu.

Diagram řešení zobrazuje způsob posílání eventu na předem nedefinovaný objekt. Třídy Button a Lever v tomto příkladu představují poskytovatele eventů (Subject viz. návrhový vzor Pozorovatel) třídy LED a Servo v tomto příkladě představují odběratele (Observer viz. návrhový vzor Pozorovatel). Třídy typu Subject reagují na podněty prostředí (zmáčknutí tlačítka, otočení kolečkem atd.) voláním funkce všech svých instancí třídy EventHolder. Volaným funkcím předávají typ eventu (stisknutí, uvolnění, překročení hodnoty atd.) a třída EventHolder poté volá na určenou instanci třídy, kterou má uloženou v paměti.

Při spuštění hlavní smyčky musí dojít k provázání Pozorovatelů (Observer) a Subjektů viz schéma 5.2. K provázání dochází voláním funkce `setEventFunction` obsažené v Interface `IEventConsumer`, ve třídě typu `Observer` (Tato třída musí dědit z `IEventConsumer`). Volané funkci se předá jako první parametr ukazatel na libovolnou funkci obsaženou v nastavované třídě. Druhý předávaný parametr je získán, jako odpověď z volání funkce `newEvent` na třídě typu `Subject` s parametrem definující, jaký typ eventu bude odchyťován.

Potřeba takovéto komunikace vznikla z důvodu, že nebylo jednoduché předem přesně určit jak bude celý program uspořádán a možné budoucí potřeby přestrukturování celého programu. Program proto potřeboval možnost navázat neznámé objekty mezi sebou a volně tím rozšiřovat program o další třídy a funkcionality. Což není v jazyce C++ snadno řešitelné.

Pro porovnání v jazyce C je takovéto volání řešeno velice jednoduše, jelikož není nutné dodržovat takové množství podmínek zajišťující „bezpečnost a přehlednost kódu“. Jazyk C nepoužívá metody, ale funkce, které nejsou pevně spjaty se žádnou třídou. Pokud se rozhodneme udržovat "třídy", předat data takovéto zjednodušené pseudotřídě, je možno jako strukturu, kterou obě funkce znají ještě před svojí inicializací.



Obrázek 5.2: Příklad spojení několika tříd návrhový vzorem Pozorovatel

Kapitola 6

Návrh a výroba desky plošného spoje

Deska plošného spoje je navržena pro připojení 16 podsvětlených tlačítek. U těchto vstupů a výstupů je z důvodu umístění do průmyslového prostředí provedena ochrana galvanickým oddělením, aby byla zachována bezpečnost desky. Deska plošného spoje obsahuje připojení ke sběrnici DALI. Dále je na této desce osazen DC/DC měnič, který převádí napájecí napětí 24 V, na potřebnou úroveň napětí pro napájení čipu mikrokontroléru 3,3 V.

6.1 Výběr mikrokontroléru

Mikrokontrolér v navrhovaném zařízení zajišťuje komunikaci po "sběrnici DALI" pomocí "DALI interface". DALI Interface převádí pouze hladinu napětí (nepřevádí na protokol UART), galvanicky odděluje vysoké napětí 16-24 V (v IDLE stavu) na napětí ve kterém je schopen operovat vybraný mikrokontrolér. Samotné vyhodnocení logických úrovní napětí komunikačního signálu a jeho převod na digitální hodnotu je poté řešen uvnitř mikrokontroléru softwarově.

6.1.1 ATMEGA328

Jednou z možností bylo použití čipu Atmega328 z důvodu dobrého obeznámení a zkušenostmi s touto platformou, velmi dobré dostupnosti, ceně čipu, a podporou bootloaderu, který zjednodušuje návrh a testování. Nepříznivé vlastnosti tohoto čipu jsou však jeho rychlost. Jelikož čip musí být schopen zpracovat komunikaci se sběrnici DALI v dostatečné rychlosti a obstarat připojené periferie (podsvícená tlačítka), tak aby nedošlo ke ztrátě informací, ze strany DALI nebo periferií.

6.1.2 STM32

Pro řešení zadání byl zvolen čip STM32 jako adekvátní náhrada čipu ATMEGA328, jelikož obsahuje větší množství vestavěných periferií. STM 32 nabízí větší počet a velikost časovačů, které jsou v čipu Atmega328 pouze 3 a jeden je dedikován pro sériovou komunikaci. Další z rozhodujících faktorů je

počet GPIO pinů, které je nutné použít pro připojení všech tlačítek a jejich podsvětlení. Dostupná vývojová deska je vybavena čipem pro sériovou komunikaci, debugování pomocí breakpointů a nahlížení do registrů zjednodušuje testování. [13]

6.1.3 Výsledný výběr mikrokontroléru

Pro finální výrobek bylo rozhodnuto využít čip STM32, a pro tento čip dále navrhnout DPS, z důvodu dostupných periférií na tomto čipu. Testování bylo prováděno paralelně na čipech (STM32, Atmega328) pro zajištění možnosti v budoucnu použít libovolný jiný čip.

6.2 Zapojení tlačítek a diod do matice

Navržený způsob zapojení periférií do matice tzn. rozdělení 16 tlačítek do skupin po 4. V každé skupině jsou prvky vodivě spojené jedním konektorem. Z každé skupiny je vytažen n-tý prvek a spojen stejným vodičem viz. obr. 6.1.

6.2.1 Zapojení tlačítek

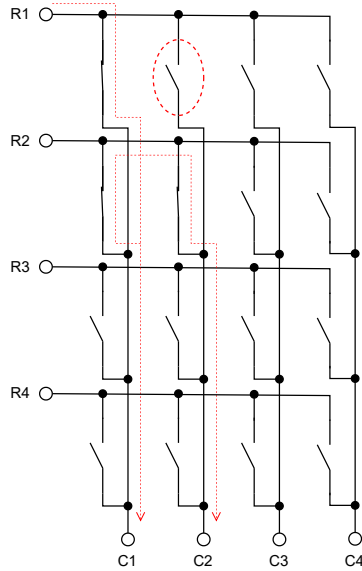
Princip vyčítání stavů tlačítek z matice je prováděn postupným propouštěním signálu do jednotlivých řad (Rx v diagramu) a sledování signálů na sloupcích (Cx v diagramu).

Toto zapojení geometrickou řadou zmenšuje počet připojených konektorů k DPS, tím se zmenšuje její velikost. V tomto uskupení tlačítek může projevit jev nazývaný Ghosting. K tomu jevu dochází při stisknutí tří a více tlačítek současně. Signál vyslaný z jednoho řádku se pak dostává na další řádky pomocí dalších stisknutých tlačítek a z těchto řádků na ostatní sloupce. V důsledku tohoto dochází k imitaci stisknutí tlačítka, které ve skutečnosti rozepnuté viz obr. 6.1

Tento problém by bylo možné odstranit přidáním sériově zapojené diody za tlačítko, čímž se zamezí zpětnému průtoku proudu ze sloupce do jiného řádku. Tlačítko s vestavěnou diodou není běžně dostupné. Řešením, by bylo z tlačítka vyvést oba vodiče do DPS a diody osadit na této desce. Tímto dojde ke zvětšení DPS, zvětšením počtu pinů a složitější montáži propojení jednotlivých tlačítek. Toto řešení není vhodné, jelikož svoji hlavní výhodu (snížení počtu konektorů na DPS) ruší. Vzhledem k možnosti Ghostingu i obtížnější montáži, toto řešení bylo zamítnuto. Pro další řešení bylo nutno přepracovat návrh na připojení každého tlačítka zvlášť. Nové řešení je popsáno v kapitole 6.4

6.2.2 Zapojení podsvícení

Ovládání matice podsvětlení je založeno na stejném principu jako u tlačítek. U tohoto zapojení nedochází k výše zmíněnému Ghostingu, jelikož podsvícení je realizováno LED diodami, které jsou již svojí konstrukcí jednocestné. Při zapojení matice dochází k jevu, kdy každá dioda svítí pouze na 25%, jelikož je rozsvícena, pouze v době zpracovávání dané skupiny, ve které je tato dioda obsažena



Obrázek 6.1: Ghosting při zapojení tlačítek do matice

(100% / 4 skupiny). Toto zapojení diod vede ke snížení světelného toku podsvícení a je třeba otestovat, zda je použitá LED dostatečně výkonná pro použití v tomto zapojení.

Po kontrole svítivosti diody na poskytnutém podsvícení, bylo zjištěno že toto podsvícení není vhodné ani pro 25% (4 skupiny) ani pro 50% (2 skupiny), jelikož podsvícení nebylo zřetelné. Z důvodu zachování dostatečné svítivosti, bylo nutné zvolit jiný návrh. Alternativní návrh bylo zapojení každé diody zvlášť. Z tohoto řešení vyplynulo 16 výstupů pro připojení diod + jeden výstup pro připojení k nulovací liště + vstup pro napájení podsvícení. Výstup na nulovací lištu bude společný s výstupem pro tlačítka.

6.3 Elektromagnetická kompatibilita EMC

Všechna elektromagnetická zařízení vyzařují elektromagnetické záření do svého okolí vlivem parazitních vlastností součástek a desky plošného spoje. Při zjišťování EMC se měří vliv elektromagnetického záření ze zařízení na okolí (EMI) a vliv okolního elektromagnetického záření na funkci elektrického obvodu (EMS). Během návrhu DPS je proto nutné dbát na zajištění obou vlivů a uzpůsobit DPS k minimalizování zmíněných rušení.

6.3.1 Elektromagnetická interference EMI

Elektromagnetická interference řeší množství a frekvenci vyzařovaného elektromagnetického záření do okolí. Každé zařízení vydává různě velké elektromagnetické záření o různých frekvencích. Dobře postavené zařízení září do okolí pouze nepatrným výkonem. Záření z plošných spojů může pocházet

i ze samotných vodivých cest mezi jednotlivými součástkami. Velký pozor je poté třeba dávat při vedení cest pro vysoké proudy, které mohou být zdrojem silného rušení. Takovéto cesty je poté třeba vést pokud možno co nejrovněji a bez vedení do pravého úhlu, které se poté chová jako cívka s velmi malým poloměrem. [14]

6.3.2 Elektromagnetická Susceptibilita EMS

Elektromagnetická susceptibilita popisuje, jaký vliv má okolní rušení na funkčnost zařízení, pokud je těmto vlivům vystaveno. Vlivy na zařízení mohou být přírodní, příkladem může být například blesk, který vydává rušení na širokém frekvenčním spektru. Za umělé vlivy lze zařadit například jiskření z motoru, datový signál z jiných zařízení nebo výkonové vedení. Dále lze vlivy rozdělit dle toho na jak širokém frekvenčním pásmu se vyskytují. Vysílače vytvářejí pro přenos dat elektromagnetické vlnění, které se ovšem indukuje i na okolních zařízeních, a mohou zařízení škodit, například chybnou interpretací stisku tlačítka, nebo narušit komunikaci s externí periferií. Proto je nutné tyto zařízení ošetřit proti vlivu takovýchto úzkopásmových rušení například pásmovým filtrem. Pokud se ovšem blízko DPS nachází jiskřiště, je na DPS přijímáno několik různých frekvencí takovýto zdroj označujeme za širokopásmový a lze se proti němu chránit například stíněním.

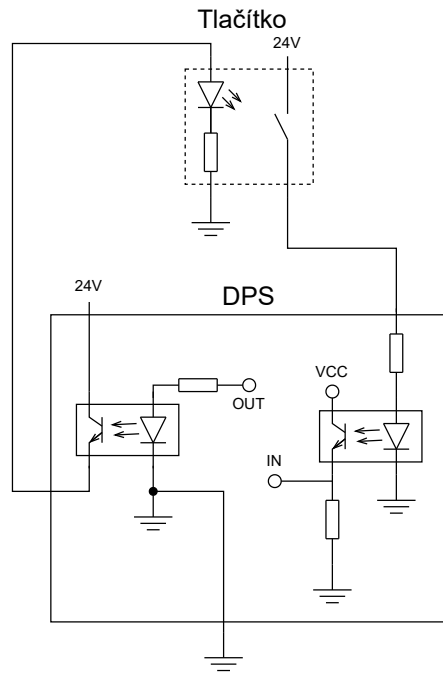
6.4 Finální zapojení DPS k podsvíceným tlačítkům

Podsvícení poskytnutých tlačítek je od výroby opatřeno rezistorem dimenzovaným přesně pro LED diodu na jmenovitou hodnotu napětí 24 V. Proto po přepracování návrhu byl stav takový, že tlačítka byla připojena k DPS šroubovacími konektory. Napětí 24 V pro napájení podsvícení řízené mikrokontrolérem je přiváděno přímo z DPS. Záporný kontakt na podsvícení tlačítka je potřeba připojit k nulovací liště mimo DPS, jelikož DPS neposkytuje z důvodu velikosti zápornou svorku pro tlačítka. Tlačítka jsou z jedné strany připojena na lištu s napájecím napětím 24 V a po stisknutí přivádí toto napětí na vstup DPS viz. obr. 6.2.

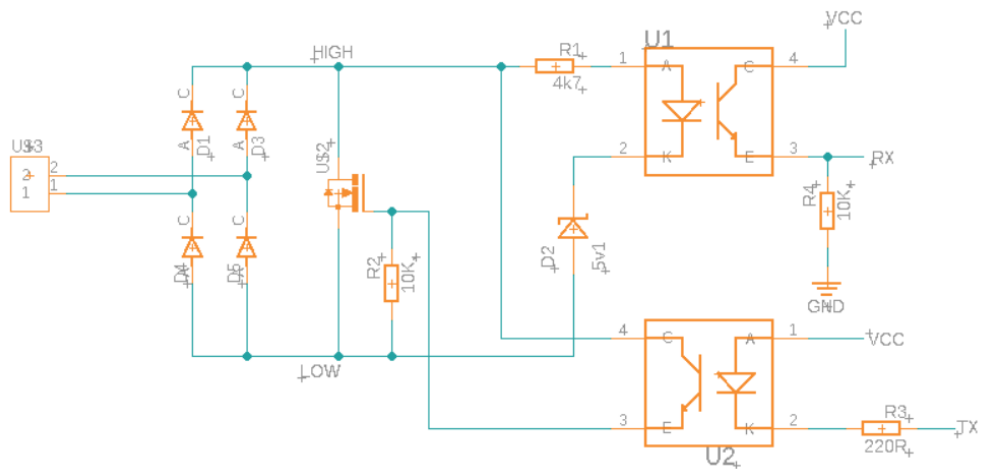
DPS je ošetřena galvanickým oddělením vytvořeným optočleny pro ochranu vstupů a výstupů viz. obr. 6.2. Pro napájení podsvícení a desky plošného spoje je vytvořen rozdílný okruh, aby bylo možné v případě potřeby změny podsvícení měnit napětí a aby nebyl čip ovlivňován touto změnou. DPS je napájena 24 V a toto napětí je dále převáděno pomocí DC/DC měniče přímo na desce plošného spoje pro odpovídající napětí, které využívá čip pro svoji logiku 3,3 V.

Rozhraní DALI je napájeno přímo ze sběrnice a řízeno přes galvanická oddělení optočleny. V prvním návrhu byl jako spínací člen využit jen MOSFET spínaný přímo napětím ze sběrnice DALI. viz obr. 6.3

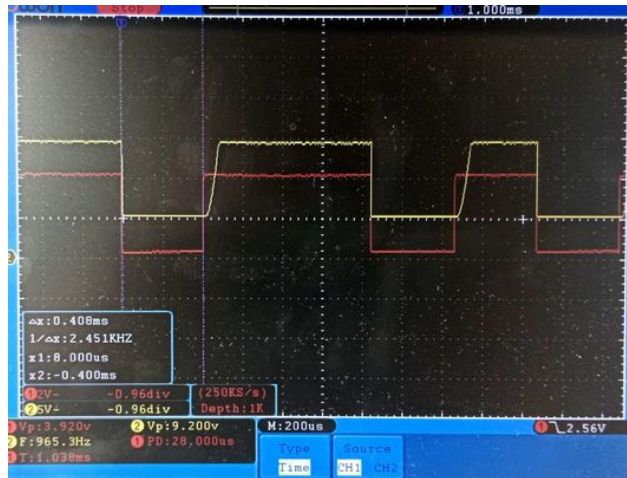
Tento návrh se ukázal jako nevhodný jelikož v důsledku klesajícího napětí na sběrnici nebylo odkud připojit napětí na Gate tranzistoru. To způsobilo, že napětí se ustálilo na přibližně 6 V viz obr. 6.4 [15]



Obrázek 6.2: Zapojení tlačítka k DPS

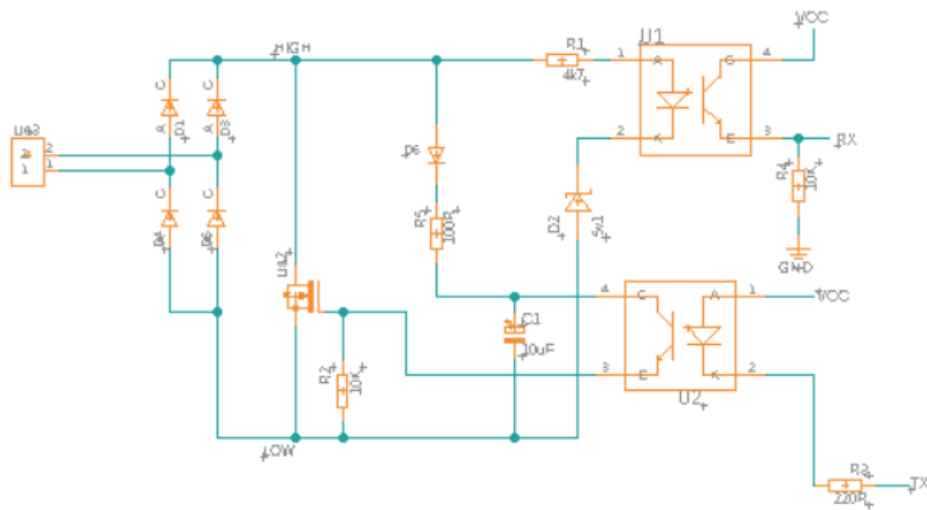


Obrázek 6.3: DALI modul V1



Obrázek 6.4: Napětí na prvním modulu CH1 - DALI CH2 - Tx

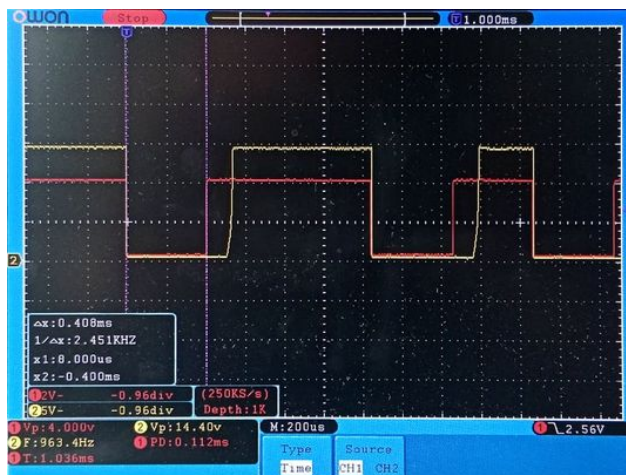
Úprava tohoto návrhu zahrnovala využití kondenzátoru, který se nabíjí, když je na sběrnici DALI vysoká hodnota. Při poklesu napětí (tj. během vysílání zprávy) poskytuje kondenzátor napětí tranzistoru. Nabíjení kondenzátoru musí být omezeno, aby nevychyloval hodnotu napětí na sběrnici dali při jeho nabíjení. Toto omezení je provedeno sériově řazeným rezistorem a diodou, která zajišťuje že se kondenzátor bude pouze nabíjet a nebude vybíjen sepnutým tranzistorem při odesílání zprávy. viz obr. 6.5 [16]



Obrázek 6.5: DALI modul V2

Po otestování tohoto modulu se napětí sice dostalo až k nule, ale odeslané zprávy byly zhruba v 80% ignorovány jako poškozené. Po důkladném proměření modulu bylo zjištěno, že tranzistor

způsobuje zpoždění ve své reakci. Tento jev byl předpokládán, ale neměl by ovlivnit komunikaci a SW byl na takovýto jev připraven. Tento jev ovšem během měření nastával jen při přechodu z nízké do vysoké hodnoty. Důvod tohoto chování bylo, že při otevření optočlenu se gate MOSFETu s téměř nulovým odporem nabil, ovšem při vybíjení stál v cestě proudu pulldown rezistor s hodnotou 10kΩ. Kapacita tranzistoru byla dostatečně velká k tomu, aby udržela otevřený přechod po dobu 150 μs. viz. obr.6.6 Po opravě změnou hodnoty rezistoru z 10 kΩ na 1 KΩ se zrychlilo vybíjení z gate tranzistoru a došlo ke vzrůstu napětí s téměř zanedbatelným zpožděním. Po této úpravě bylo při testu odesílání téměř 100% úspěšné.

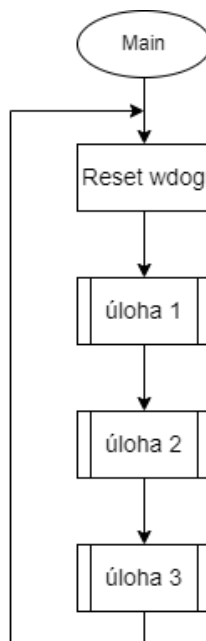


Obrázek 6.6: Napětí na druhém modulu CH1 - DALI CH2 - Tx

Kapitola 7

Softwarové řešení diplomové práce

Diplomová práce bude z důvodu rozsáhlosti řešena v "úlohách". Jedna z variant řešení práce je, že úlohy budou spouštěny po sobě (bez použití operačního systému) viz. obr. 7.1 z tohoto důvodu je nutné použít watchdog, který zajistí aby při narušení chodu programu nedošlo k "zatuhnutí" ale při detekci problému provede restart mikrokontroléru. Tímto se zajistí nenarušení sběrnice DALI ze strany chyby mikrokontroléru. Nevýhodu v podobě snížené rychlosti vyvolané opakovaným voláním funkcí vyváží přehlednost držných dat a volaných akcí na mikrokontroléru v přesně stanoveném pořadí. V případě rozšiřování a použití operačního systému reálného času v budoucnu tím bude vše připraveno pro přechod, který tím pádem proběhne mnohem snáze a rychleji.



Obrázek 7.1: Stavový diagram sériového zpracování úloh

V hlavní smyčce programu se tím pádem bude odehrávat i předávání dat mezi úlohami, což by v případě programování v prostředí s operačním systémem zajišťoval operační systém, proto je při programování na toto nutno brát zřetel. Pro zajištění kontinuálního běhu programu nesmí žádná z úloh obsahovat nekonečnou smyčku, nebo zastavení programu.

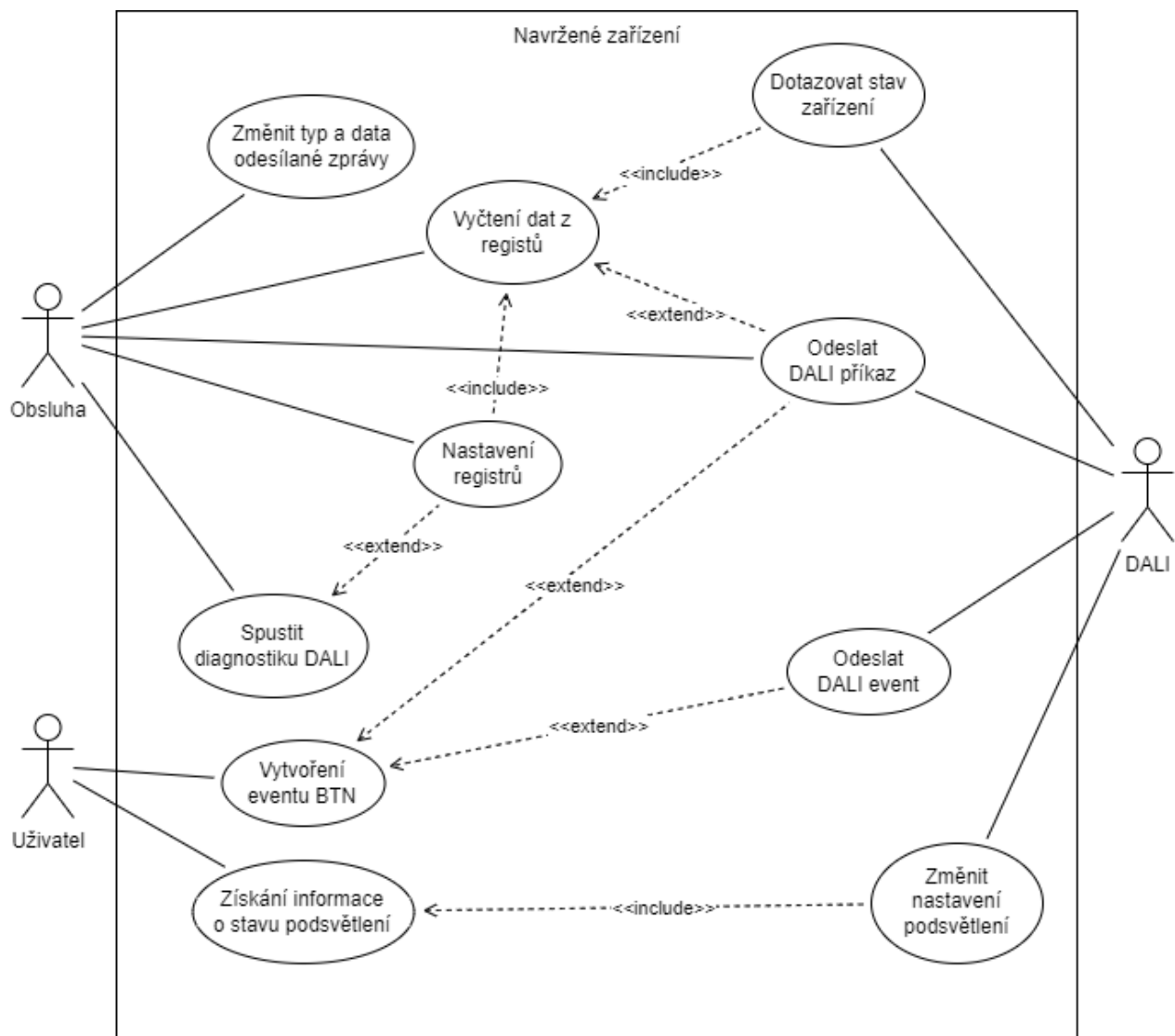
7.1 Přehled možných způsobů komunikace zařízení

Pro komunikaci s poskytnutým DALI modulem je požadována možnost komunikovat se zařízením po sběrnici DALI ne jen pomocí 24-bit rámce, který udává eventy, ale i vyčítat stavy tlačítek periodicky pomocí 16-bit rámce. Z tohoto důvodu se proto nemůže jednat o původně zamýšlené zařízení typu input device viz. 2.2.2. Je tedy nutno zařízení programovat jako aplikační kontrolér viz. 2.2.3. Takto bude možno naprogramované zařízení kontaktovat jak 24-bit rámcem tak 16-bit rámcem a navíc bude moci zařízení přímo přistupovat ke zvoleným světlům. Na vypracování této práce, to sice nemá žádný vliv, protože by řídicí uzel neměl informaci o takto vyvolaných změnách osvětlení, ale tato možnost zůstane zachována pro možnost budoucího rozšíření. Změnou na aplikační kontrolér zároveň dostane zařízení novou možnost připojení k PC pomocí programovacích portů. Těmito porty bude možno zařízení pohodlně programovat a kontrolovat dění na sběrnici, čímž dojde ke značnému zjednodušení z pohledu vývoje a ladění při montáži zařízení.

Diagram 7.2 vypovídá o tom, jaké možnosti má uživatel, který není připojen pomocí programovacího portu, ale má k dispozici pouze poskytnutá tlačítka na zařízení oproti obsluze zařízení, kterou představuje programátor nebo seřizovač při montáži, díky připojení programovacím portem. Na diagramu je dále vidět, ke kterým možnostem se dostanou ostatní zařízení připojená na sběrnici DALI.

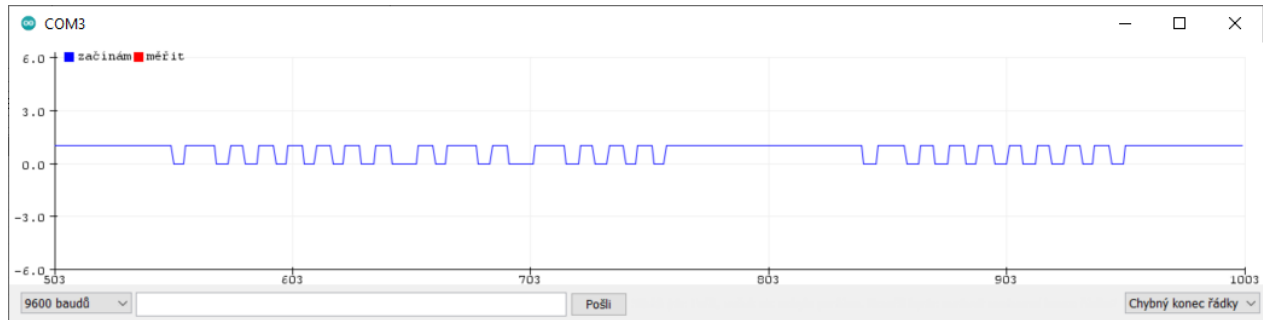
Pro diagnostiku DALI byly navrženy tři možnosti, pro různé úrovně nahlížení na komunikaci běžící na sběrnici. Diagnostika se provádí za připojení programátoru k desce plošného spoje, a je přenášena po sériové lince do počítače. V počítači je možno pomocí různých programů třetí strany provádět čtení těchto zpráv. Diagnostika s nejvyšší úrovní abstrakce je výpis přijatých zpráv, kdy každá přijatá zpráva je vypsána přes sériovou linku jako jedničky a nuly zakódované v ASCII kódu. Mezi jednotlivými zprávami je symbol znaku konce řádku (char: '\n' nebo v ASCII: 0x0D). V případě, že by zařízení nebylo schopno rozlišit jedničku nebo nulu, odešle symbol odpovídající této chybě. Seznam a významy těchto chyb je možno nalézt v kapitole dále viz. 7.3.2.

Druhou variantou vypisování zpráv je odeslání seznamu času eventů (změn hodnot napětí na sběrnici DALI), ke kterým došlo během jedné zprávy. Toto nastavení by mělo pomoci při hledání chyby v časování, popřípadě v důsledku špatného čtení zapříčiněného kapacitou sběrnice DALI. Třetí funkce byla vytvořena pro hledání chyb při kolizích, nebo špatném vysílání na tranzistoru vysílače. Tato funkce odesílá zprávy uzpůsobené pro software Arduino ploter, který umí vykreslit graf ze získaných hodnot viz. obr. 7.3. Takto může obsluha vidět všechny zprávy graficky. Opticky tak rychle odhalí chybu způsobenou třeba rušením, nebo vypadnutím spojení s tranzistorem. Z



Obrázek 7.2: Use case diagram navrhovaného zařízení

důvodu, že uchovávání hodnot času je velmi náročné na paměť, jsou tyto dva režimy omezeny počtem zpracovávaných bodů.



Obrázek 7.3: Ukázka vykreslení zprávy programem třetí strany (arduinoIDE-Sériový monitor)

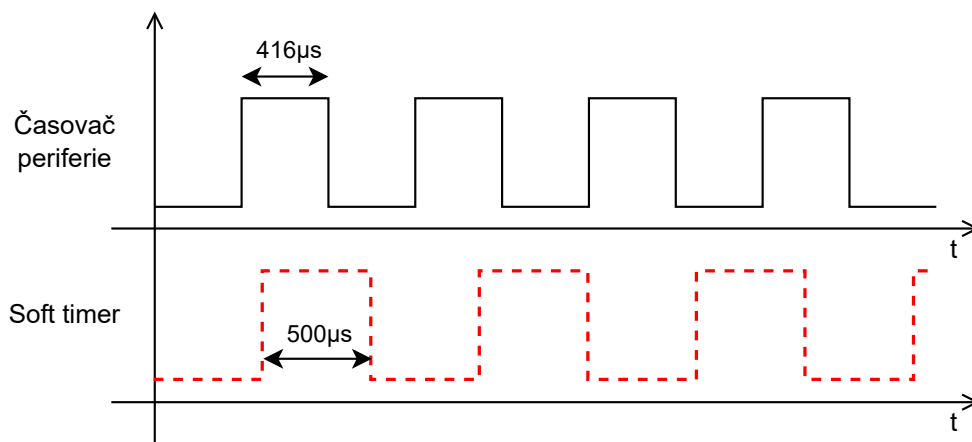
7.2 Využití přerušení pro řízení chodu programu

Alternativou k použití sériového zpracování dat tak, jak je tomu popsáno v kapitole 7, je zpracovávat data na základě impulsu (přerušení) z periferií (přerušení ze vstupu IO portu, nebo časovače). Tato metoda nevytěžuje tolik procesor mikrokontroléru, ovšem je složitější pro ladění, jelikož není zaručeno pořadí zpracování. [9]

Vzhledem k tomu, že během testování se projevila dilatace časového měřítka způsobená použitím soft timeru pro zajištění kontinuity běhu programu, viz obr. 7.4 bylo nejvhodnější zkombinovat tyto metody: metodu cyklického dotazování (pollingu) pro tlačítka, protože zpracování tohoto signálu není tolik kriticky závislé na čase a nekomplikuje zbytečně běh programu. Pokud by byla použita metoda reakce na přerušení, bylo by nutné zajistit, aby nedocházelo k překrývání přerušení a špatnému vyčítání, nebo zápisu hodnot v místech programu, kde na to program není připraven. Reakci na přerušení bylo naopak vhodné využít u zpracování komunikace s DALI, které je kriticky závislá na čase. Software je nutno upravit tak, aby předpokládal více možných vstupů přerušení a i při překrytí těchto přerušení byl schopen vykonávat svoji funkci správně.

Přechodem na použití kombinace pollingu a reakcí na přerušení vznikl program, pro jehož představu je vyobrazen diagram viz. 7.5. V tomto diagramu jsou vidět celkem tři "vlákna", která řídí celé zařízení. Ve skutečnosti je zde ještě čtvrté vlákno, které tiká periodicky stále od spuštění mikrokontroléru a počítá čas, podle kterého se vlákno pro čtení orientuje.

Přerušení Timer send se odesílá periodicky s periodou 416 μ s což je polovina času jednoho bitu. Při každém impulsu vyslaném tímto časovačem může na základě přenášené informace provést překlopení hodnoty napětí na sběrnici DALI. Seznam těchto hodnot je generován z požadované zprávy a generuje se při požadavku na jeho odeslání. Po dokončení odeslání všech změn časovač ukončuje odesílání přerušení. Spuštění přerušení a čtení tohoto časovače je uskutečněno při požadavku na novou zprávu.



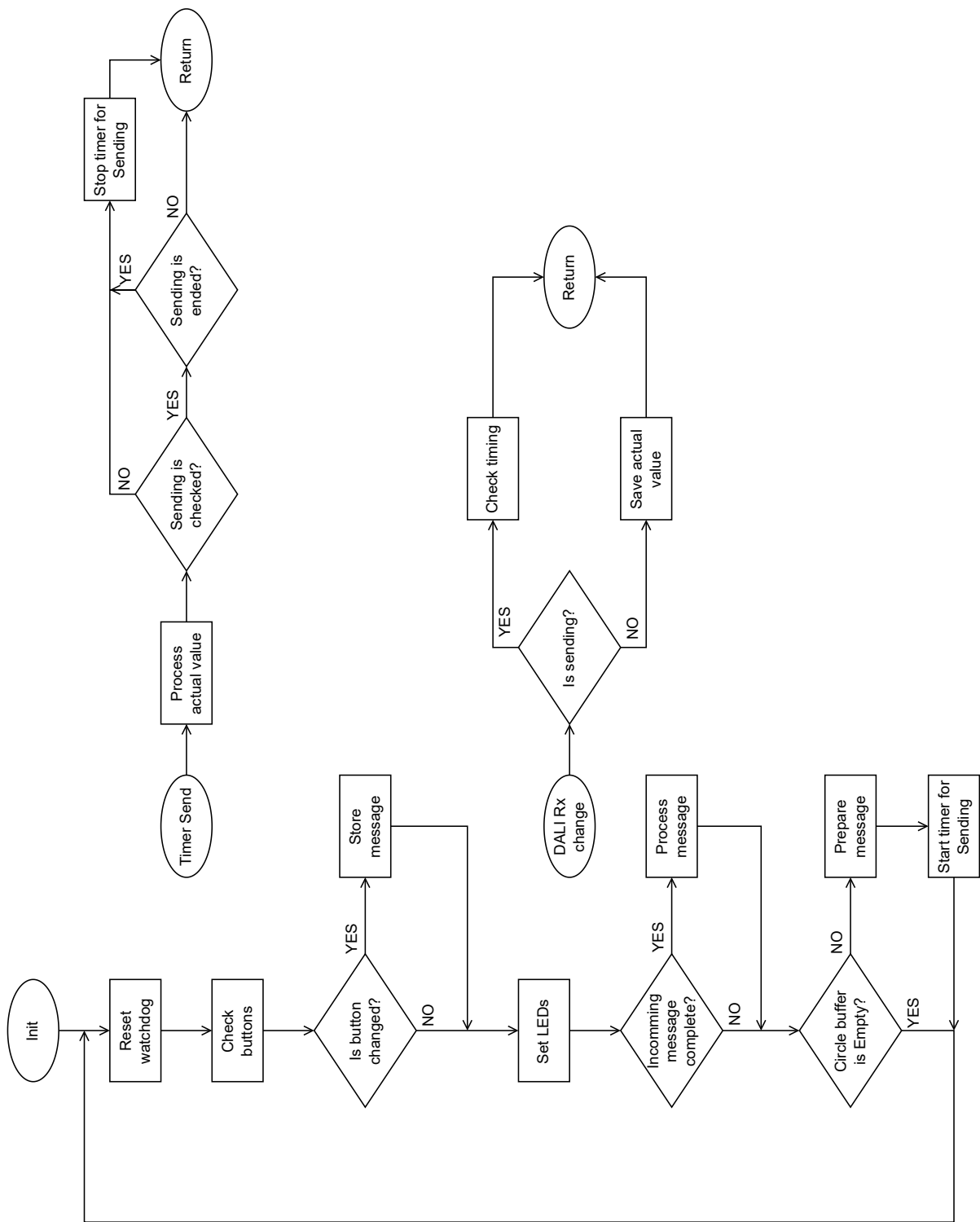
Obrázek 7.4: Problém dilatace při použití soft timer

Dalším přerušením, řídicí chod programu, je přerušení z pinu Rx v rozhraní DALI. Po přechodu programu do tohoto přerušení jsou dále řešeny dva režimy. V režimu odesílání zprávy je kontrolováno, zda ke změně došlo v čase stanoveném jako povolená odchylka od odeslání požadavku na změnu. Druhý režim slouží pro přijetí zpráv ze sběrnice DALI a při přijetí se uloží aktuální čas z časovače pro čtení a logická hodnota, na kterou se napětí na sběrnici DALI změnilo.

Pro jednodušší zpracování celého programu bude dále v této práci popisován celý program objektově, přesto, že zvolený jazyk pro programování C, není objektově orientovaný programovací jazyk. K docílení určitého objektového rozložení, může být využito zapouzdření všech proměnných takovéto pseudotřídy do struktury a při volání funkce, respektive pseudometody, poté jako první parametr vložit ukazatel na patřičnou strukturu. Využitím rozparcelováním na objekty dojde ke zrychlení předávání velkého množství parametrů a zkrátí se tak celý psaný program, který navíc získá přehledné uspořádání pro budoucí čtenáře.

Pokud není potřeba disponovat větším množstvím instancí dané třídy je možné třídu implementovat do souborů header file (.h) a zde poskytnout klíčovým slovem extern všechny proměnné které jsou typu public. Dále jsou zde uvedeny i public funkce (pseudometody). Proměnné typu public i private jsou vytvořeny jako globální proměnné uvnitř source file (.c). V tomto dokumentu jsou uvedeny také lokální funkce a implementace header file, čímž se získá reference pro globální funkce pro tuto třídu. Bohužel není žádný způsob jak rozdělit namespace těchto dokumentů jelikož jazyk C nic takového neimplementuje. Řešení je zavedení prefixů. V případě tohoto projektu jsou prefixy zvoleny dle třídy ve které se nacházení pro příklad proměnná var, nacházející se ve třídě Class bude označována jako Class_var.

Program je možno za výše zmíněných okolností popsat class diagramem viz. 7.6. Na tomto diagramu je možno vidět způsob rozdělení řešení komunikace na dvě hlavní třídy, DALI_Send a DALI_Receive. Třída DALI_Send obsahuje stavové automaty pro řešení odesílání a kontrolu



Obrázek 7.5: Řešení programu kombinací pollingu a reakcí na přerušení

správnosti odeslání zprávy. Třída DALI_Receive rozhoduje, zda přijatý signál dále pře pošle třídě DALI_Send, nebo zda uloží změnu napětí na sběrnici DALI. K rozhodnutí dochází podle toho, zda zařízení je ve stavu odesílání zprávy. Dále tato třída disponuje funkcí pro zpracování nasbíraných změn a překlad na zprávu dle přijatých parametrů.

Pro zajištění a zakonzervování všech funkcionalit byla vytvořena třída DALI, která by jako jediná měla být volána z ostatních tříd a z hlavní smyčky. Tímto je docíleno zapouzdření a kód je připraven pro napojení na další třídy a funkce, starající se o ostatní funkcionality, které musí mikrokontrolér provádět. Ve třídě DALI je uchovávan také seznam všech změn napětí, jejich časů a hodnot. Pro uživatele je připraven i ladící mód, kdy předáním ukazatele na neblokující funkci může vyžádat zavolání při přijetí nové zprávy, popřípadě po jejím dekodování a neschopnosti automaticky reagovat na neznámý nebo nedefinovaný příkaz.

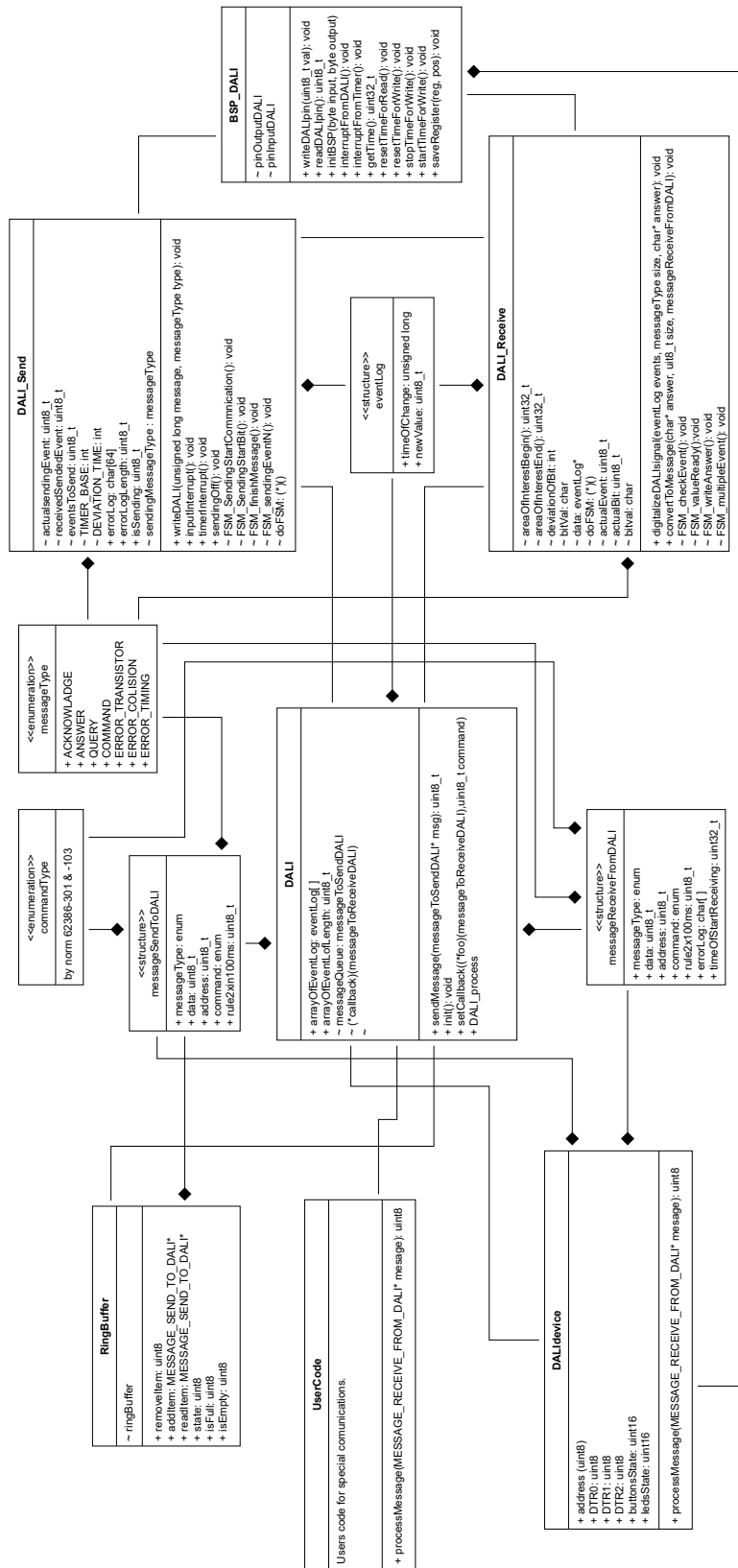
Celý program by měl být přenosný na další platformy v případě potřeby její změny, což není možné, pokud by některá z vnitřních tříd reagovala, nebo zapisovala na jakýkoliv z fyzických registů čipu, nebo jeho periferií. Z tohoto důvodu byly všechny volání na periferie čipu přesunuty do jediné třídy BSP_DALI, která je při změně platformy jednoduše upravitelná pro vlastnosti nové platformy. Tímto se zamezí vstupování do kódu, který již jednou byl testován a možnosti vytvoření chyby, která by nemusela být dohledatelná.

Z důvodu velkého množství tlačítek je možné, že bude požadováno odeslání více zpráv, nebo budou požadavky zadány rychleji, než stačí zařízení odesílat. Aby nedocházelo ke ztrátě dat byl vytvořen kruhový buffer, který zajišťuje ukládání požadavků na odeslání. Kruhový buffer není časově tolik nákladný na správu, jelikož se data v něm uložená nemusí pohybovat. Třída kruhového bufferu RingBuffer poskytuje možnost přidat nový prvek, kdy zkontroluje, zda je možné prvek přidat (buffer není přeplněný) a poté vrátí ukazatel na pozici, kde je nový prvek uložen. Dalšími poskytovanými funkcemi je vyčtení, které opět vrátí ukazatel na pozici aktuálního prvku pro čtení, ale nevymaže jej, dokud nebude zavolán příkaz remove. Tímto je zajištěna možnost vyčítat vícekrát a v případě potřeby opakovat odeslání kódu.

DALI_device je třída, ve které jsou uloženy registry, kterými musí zařízení disponovat, reakce na zprávy, které jsou předem určeny normou pro patřičný typ zařízení a je zde uložena zpráva, která očekává druhé potvrzovací volání.

Struktury zobrazené v diagramu jsou použity pro zjednodušení práce se zprávami, kdy je možno s daty pracovat najednou, místo odesílání funkcí s několika parametry. Tento koncept zjednodušuje čas při předávání a práci se zprávami. Enumerátory zastupují pouze číselné vyjádření typů dat dle normy, a zjednodušují programování.

Hlavní smyčka běží ve třídě user code a je ponechána pro změny týkající se funkčnosti zařízení, obsluhou podsvícení, čtení hodnot z tlačítek a dalších uživatelem potřebných neblokujících funkcí.

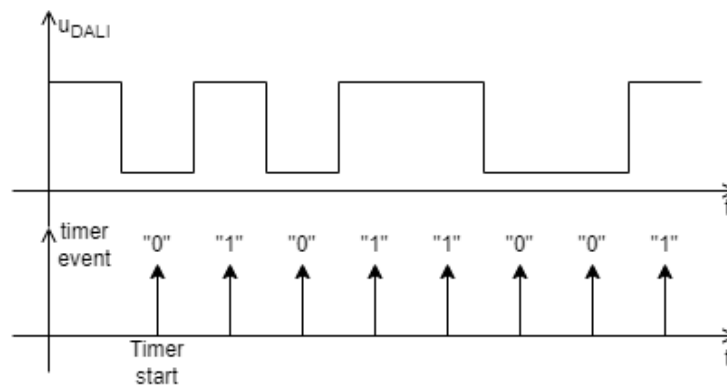


Obrázek 7.6: Diagram tříd DALI knihovny

7.3 komunikace po sběrnici DALI

Komunikace po sběrnici je prováděna dvěma piny, první pin Rx pro čtení z rozhraní DALI, tento pin je nastaven jako input. Druhý pin je Tx určen pro zápis na sběrnici DALI, skrze rozhraní. Tx mění svoje nastavení z output low pro logickou "0" na high impedance pro log "1". Softwarově je komunikace řešena jako sada stavových automatů spouštěných ze tří zdrojů signálu viz. obr. 7.5.

Ze sběrnice je možno vyčítat různými způsoby jedním ze způsobů čtení ze sběrnice DALI je nastavit časovač na pevnou frekvenci odpovídající polovině času jednoho bitu a započít čtení od poloviny této frekvence. V Oblastech čtení by poté měly být hodnoty stabilní a nemělo by docházet k přechodovým jevům. viz obr. 7.7. U metody zachytávání hodnot časovačem se ovšem ztrácí základní výhoda kódování manchester, což je možnost zpětné kalibrace při čtení, za použití přechodů mezi úrovněmi napětí. Proto není tato metoda vhodná pro použití u této práce. [17]

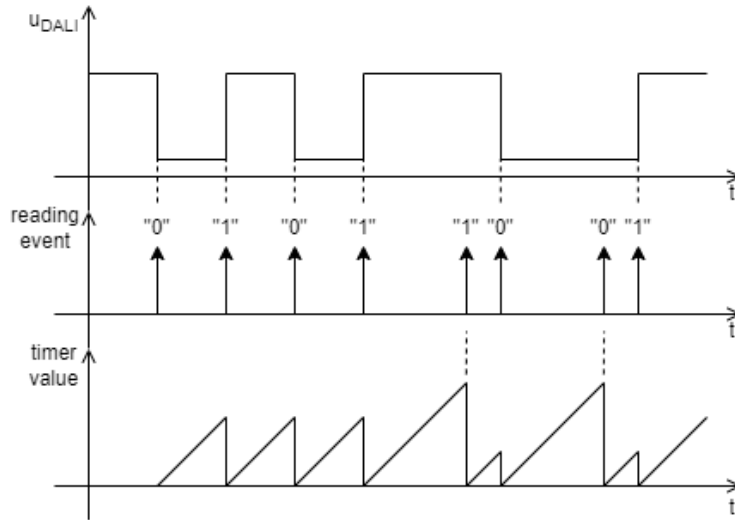


Obrázek 7.7: Metoda čtení signálu manchester časovačem

Druhá metoda zaměřená na zpětnou kalibraci časování při čtení, je využití časovače a přerušení ze vstupu. Kombinací časovače a přerušení ze vstupu je možno startovat časovač, vždy při přechodu mezi úrovněmi napětí. viz obr. 7.8. Tato metoda je ovšem složitá v porovnání, kolik možností přináší. Z tohoto důvodu není vhodná pro řešení této diplomové práce.

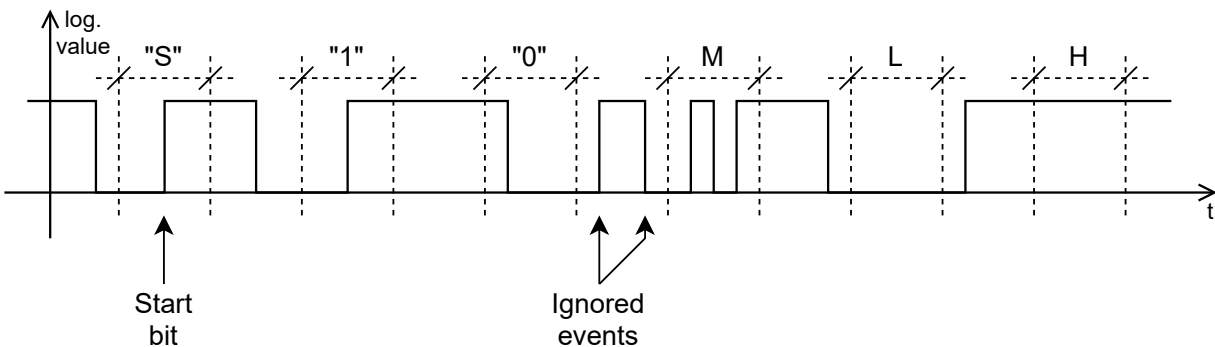
Třetí metoda je v porovnání s předchozími metodami zřejmě jednodušší pro zachytávání. Využívá pouze přerušení ze vstupního pinu Rx. Při zachycení změny napětí na sběrnici DALI uloží aktuální hodnotu času, který je generován přiřazeným časovačem. Po přijetí celé zprávy je možno vyhodnotit přijaté změny hodnoty napětí pomocí různých matematických operací a zajistit tak nejen synchronizaci čtení, jak je tomu u předchozí metody, ale i například odrušení popřípadě práci s kapacitou sběrnice.

Při komunikaci se na sběrnici může vyskytnout několik různých chyb, se kterými je nutno pracovat a umět je vyhodnotit. Příčin chyb může být celá řada, ale je možno je rozdělit do tří základních skupin. První skupinou je chyba rušením viz. 7.9. K chybě rušením může nastat při vystavení sběrnice střídavému elektromagnetickému poli. Tato chyba se projeví větším množstvím změn v době



Obrázek 7.8: Metoda čtení manchester kombinací časovače a přerušení z pinu Rx

jednoho bitu. Další chybou je zkratování sběrnice. Tato chyba může nastat při poškození vysílajícího tranzistoru, nebo může být zapříčiněna kapacitou na sběrnici. V některých případech se nemusí jednat o chybu, ale delší setrvání ve zkratovaném stavu má význam zprávy pro restart napájení. Posledním typem chyby je setrvání ve stavu vysoké logické úrovně, který může být zapříčiněn poškozeným vysílajícím tranzistorem, nebo může mít význam konce zprávy.



Obrázek 7.9: Graf rozklíčování signálu

7.3.1 Příjem zpráv po sběrnici DALI

Zařízení neobsahuje jen vstupní periferie v podobě tlačítek, ale i výstupní periferii v podobě podsvětlení. Tím vzniká potřeba zprávy i přijímat. Tyto zprávy pocházejí z nadřizovaného uzlu (PLC) a měly by obsahovat informace o zapnutí/vypnutí jednotlivých podsvětlení tlačítek. Úloha pro čtení dat ze sběrnice bude spouštěna ne hlavní smyčkou ale pomocí přerušení závislého na změně stavu

na pinu Rx rozhraní sběrnice DALI. Pokud není zařízení ve stavu odesílání zprávy, úloha uloží data o času přijetí změny a stavu, který se nově projevil na sběrnici viz. 7.1. Po přijetí stop bitu se spustí řetězec diagnostiky přijaté zprávy. V prvním bodě se převede seznam digitální hodnoty reprezentovanou polem znaků typu char (uint8_t) které mohou nabývat hodnot viz. tab. 7.1

Tabulka 7.1: Značení možných přijatých stavů bitu

| States of received bit | |
|------------------------|--------------------|
| '0' | Logical 0 |
| '1' | Logical 1 |
| 'M' | Multiple event |
| 'H' | High voltage level |
| 'L' | Low voltage level |

Pokud přeložená zpráva obsahuje pouze znaky '1' a '0' dále se zpracovává a překládá do struktury, ve které je uložena adresa, typ zprávy, příkaz pro zpracování, nesená data, popřípadě adresa instance ze které byla zpráva odeslána. Takto zpracovaná zpráva se dále odesílá ke zpracování do třídy DALIdevice, kde se zpracuje, pokud je tento příkaz definován normou. V opačném případě je zpráva odeslána do uživatelského kódu, nebo ignorována pokud uživatelský kód není implementován.

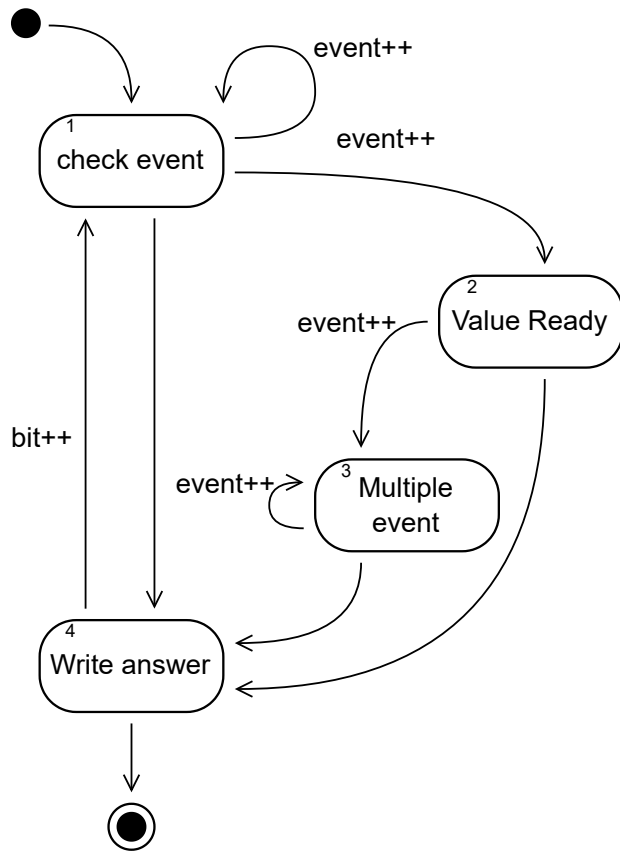
7.3.2 Zpracování eventů do symboliky dat

Zpracování přijatých eventů (změn napětí na sběrnici DALI) se provádí stavovým automatem. Stavový automat je spouštěn při kompletním přijetí zprávy a jeho provádění se opakuje ve smyčce, dokud postupně nezpracuje všechny přijaté změny napětí viz. 7.10. Po dokončení překlada všech změn napětí dojde ke kontrole zda poslední změna napětí patří poslednímu bitu, pro zapsání hodnoty '1', která v takovém případě není odhalena stavovým automatem.

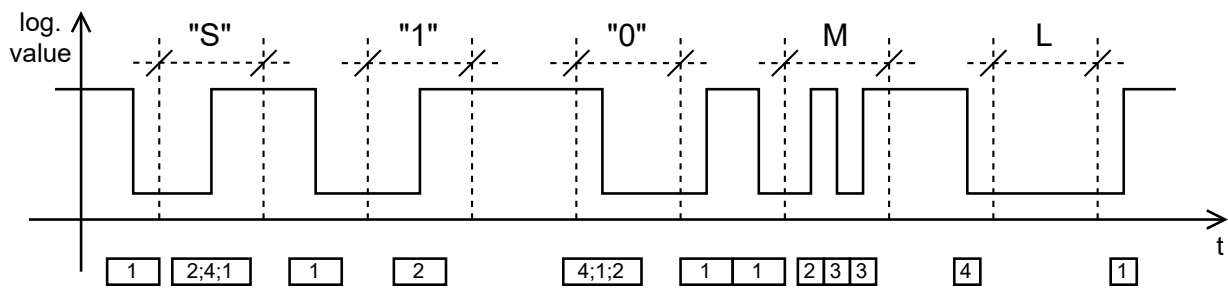
Ve stavu Check event je zjišťováno, zda zkoumaný event patří do času pro překlopení bitu. Pokud tento čas odpovídá požadovanému času pro bit, stavový automat se překloupí do stavu Value ready zde je zkontrolováno, že aktuální event není v požadovaném pásmu spolu s dalšími jak je tomu v přehledu viz. 7.11 v tomto přehledu je možno vidět stavy, které se spouští pro daný event. a jaké označení bitu z nich vyplývá. označení se vždy přiřazuje až ve stavu Write answer, odkud se automaticky vrací zpět do prvního stavu Check event. Na obrázku 7.11 je také vidět stav stavového automatu viz. obr. 7.10, kterými tento stavový automat prochází.

7.3.3 Zápis zprávy

Při zápisu zprávy je zpráva zpětně kontrolována. Během přenosu mezi čipem a sběrnici DALI je předpokládáno zpoždění, které ovšem může nabývat pouze daného času viz. 7.12. chyby mohou být způsobeny konfliktem odesílaných zpráv, pokud dvě zařízení začnou odesílat najednou, dojde k tomu, že jedno ze zařízení bude očekávat vysokou úroveň na sběrnici, ale druhé zařízení zkratuje sběrnici

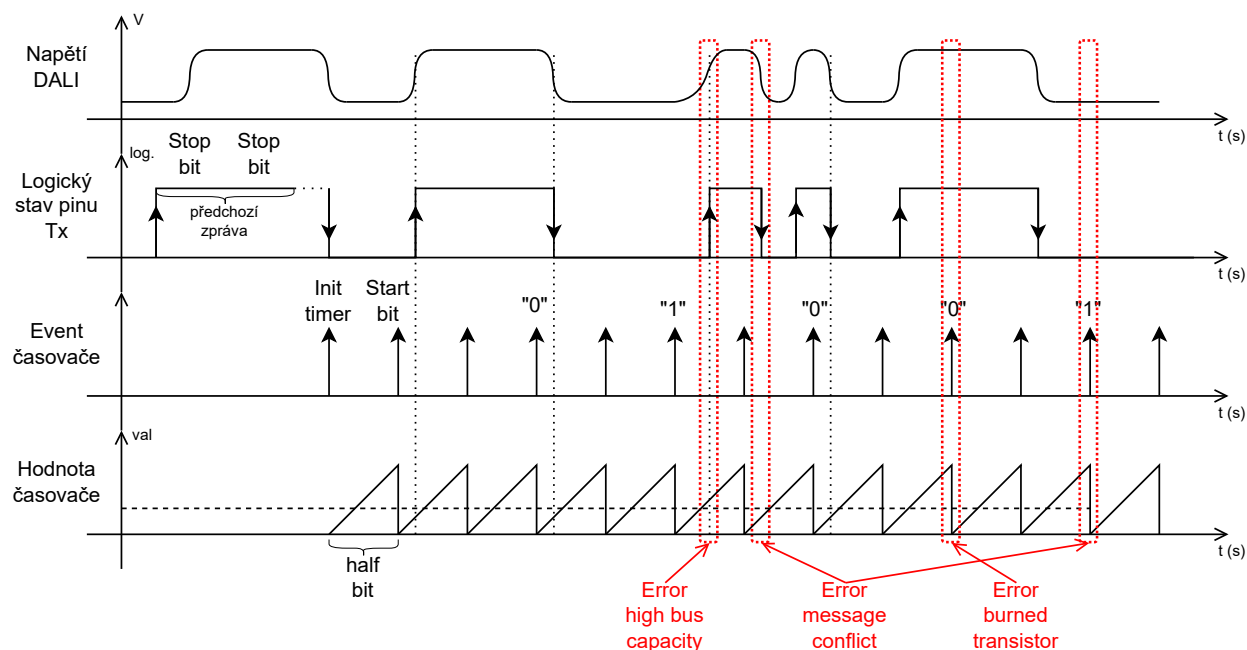


Obrázek 7.10: FSM Diagram rozklíčování signálu



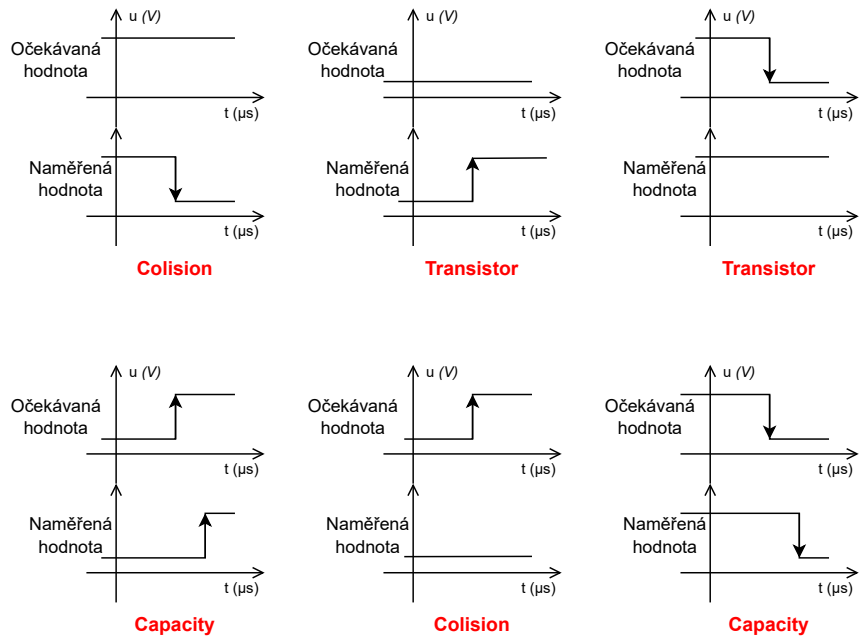
Obrázek 7.11: Rozklíčování signálu FSM časový průběh

aby vytvořilo logickou nulu. Další chybou může být vadný tranzistor, který není schopen odeslat logickou hodnotu nula. Při opakování této chyby by se mělo zařízení odstranit aby nevytvářelo konflikty a chyby při pokusech odeslat zprávu s poškozeným tranzistorem. chybu tranzistoru může totiž zapříčinit i kondenzátor, který při nedodání dostatečného množství energie dostane napětí na sběrnici do zakázaného stavu mezi logickými úrovněmi napětí. Třetí možnou chybou je vysoká kapacita sběrnice, u které dochází ke zkreslení tvaru signálu a znemožňuje přeposílání zpráv. Úplný přehled typů chyb je vidět na obr. 7.13 a obr. 7.12

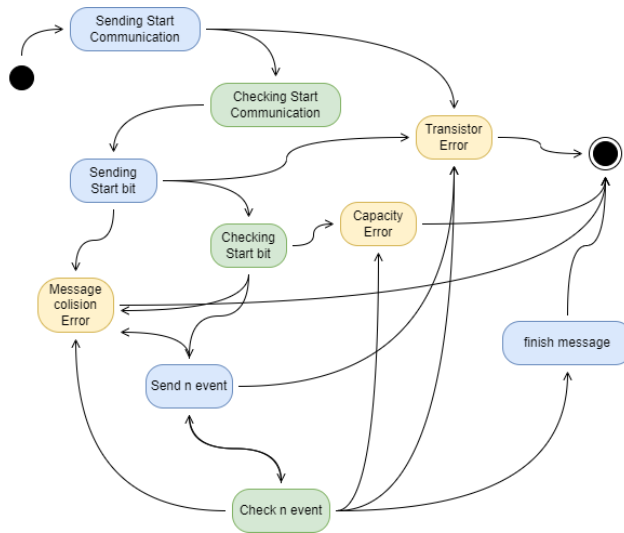


Obrázek 7.12: Chyby při odesílání dat na sběrnici

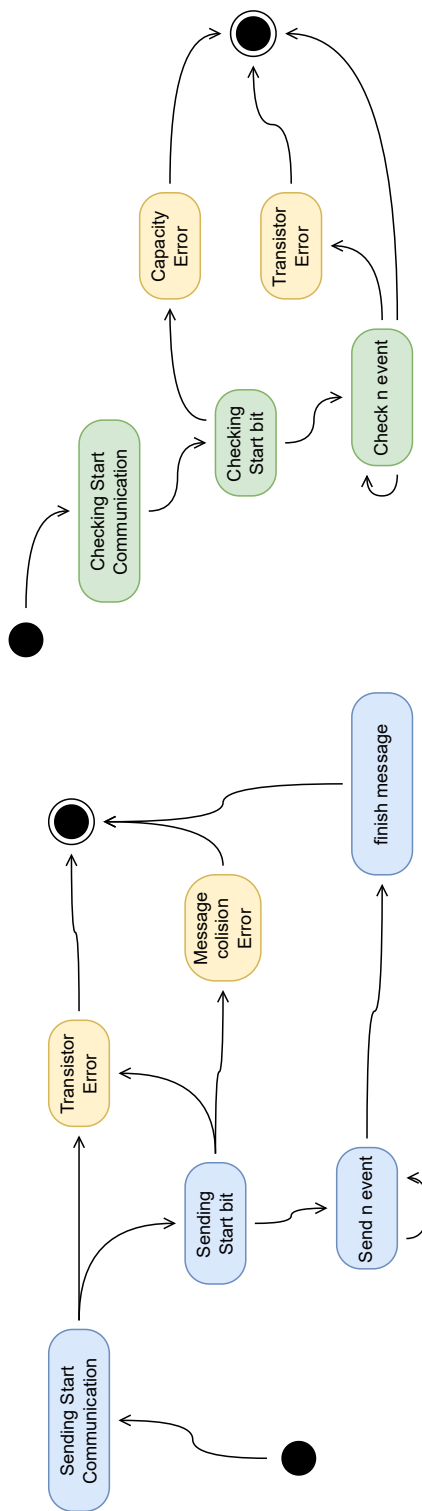
Odesílání se řídí stavovým automatem popsaným na obr.4.1. Tento stavový automat ovšem není možné implementovat, jelikož reaguje a popisuje dva různé zdroje signálů. Dle tohoto diagramu je ovšem zřetelné, že jakákoliv chyba musí okamžitě vest k zastavení odesílání aktuální zprávy. Dále je na diagramu patrné, že se střídá využívání přerušování z časovače a ze vstupního pinu Rx. Pro implemetaci je poté vhodné využít diagramy odděleně viz. diagramy 7.15



Obrázek 7.13: Typy chyb na sběrnici při odesílání



Obrázek 7.14: Stavový diagram pro odesílání zprávy



Obrázek 7.15: Stavové automaty pro odesílání a kontrolu zprávy upravené pro implementaci

Kapitola 8

Ověření výsledků

Celá práce byla testována na poskytnuté sběrnici firmou Ingeteam. Byly testovány všechny 3 metody přenosu informací (přímé řízení osvětlení tlačítka, odesílání eventů, periodické dotazování řídicího uzlu) i oba možné komunikační protokoly (24-bit, 16-bit). Úspěšně se podařilo ovládat dostupné osvětlení pomocí tlačítek připojenými k navrženému zařízení. Zařízení bylo úspěšně schopno přijmout jakoukoliv zprávu z hlavního uzlu a odpovědět na jeho požadavky (zprávy typu "Query").

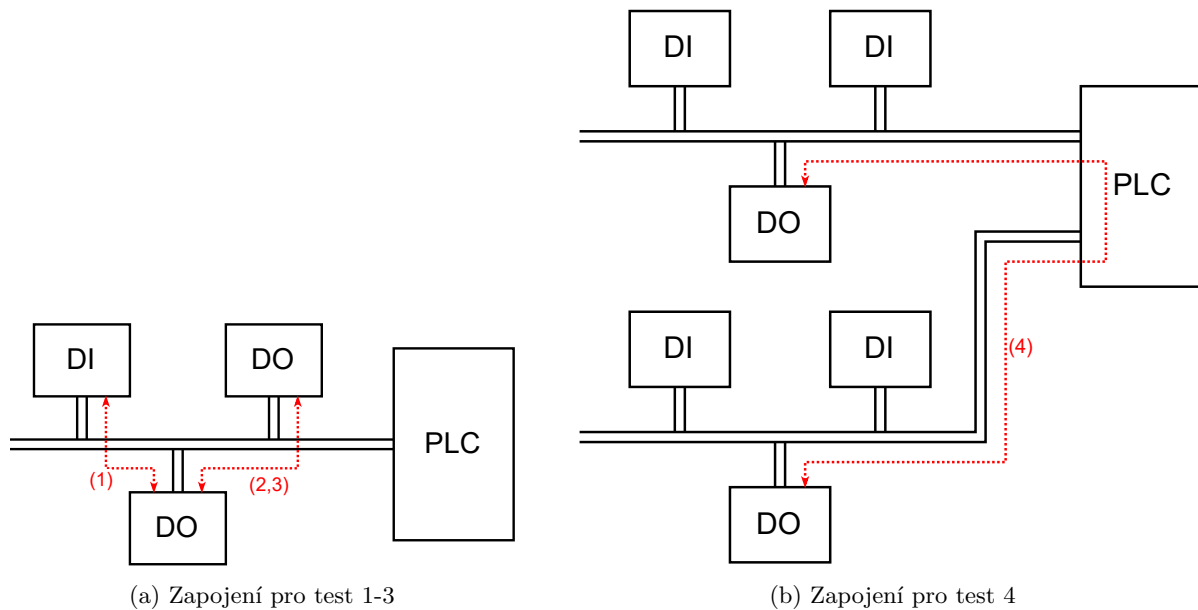
8.1 Testovaná zapojení

Zařízení bylo testováno na připojení ve stejné větvi viz. obr. 8.1a, kde byl prováděn první test na komunikaci s osvětlením a testy na rychlost komunikace mezi jednotlivými body na sběrnici. Rychlost komunikace může být ovlivněna zejména pořadím odesílání, při kterém má navržené zařízení nižší prioritu odeslání než hlavní uzel. Jelikož poskytnutý hlavní uzel používá jiný časový protokol než navržené zařízení. Navržené zařízení musí vyčkat, než hlavní uzel vyřídí veškerou komunikaci kontrolující stav osvětlení a nechá komunikovat ostatní zařízení připojená na sběrnici DALI.

Druhým zapojením viz. 8.1b bylo možno ověřit pouze komunikaci formou polling (dotazování), jelikož poskytnutý hlavní uzel nepodporuje 24 bitový frame a tudíž ani není schopen přijmout a rozpoznat zprávu typu "event".

8.2 Vyhodnocení naměřených hodnot

Hodnoty byly měřeny osciloskopem jako rozdíl mezi hranou napětí při stisknutí tlačítka, a hranou napětí rozsvícení podsvícení tlačítka. při měření na stejné větvi bylo měření prováděno na jednom zařízení a druhé sloužilo jako bod pro odpověď místo uzlu, který není schopen event přijmout. V případě měření na dvou větvích bylo provedeno čtení tlačítka u jedné větve a měření podsvícení v druhé větvi. Hodnoty byly vyneseny v tabulkách viz. 8.1 a 8.2.



Obrázek 8.1: Využitá zapojení pro měření

Tabulka 8.1: Zpoždění bez komunikace ostatních zařízení

| * | Event | Polling |
|--------------|-----------|-------------|
| stejná větev | 53ms-55ms | 145ms-200ms |
| jiná větev | - | 145ms-200ms |

Tabulka 8.2: Zpoždění s vlivem komunikace ostatních zařízení

| * | Event | Polling |
|--------------|-----------|-------------|
| stejná větev | 53ms-90ms | 145ms - >2s |
| jiná větev | - | 145ms - >2s |

Z tabulek je vidět, že hodnoty zpoždění mezi stisknutím tlačítka a obdržení zpětné vazby je zásadně nižší u zpráv posílané jako event. Hlavním důvodem, je zpoždění mezi jednotlivými dotazy, které vzniká na modulu PLC poskytnutého uzlu. Z tohoto důvodu je také zvýhodněná metoda posílání eventů v tabulce 8.2, jelikož se odpověď stačí odeslat dříve, než se zpracuje zpráva v modulu PLC. Při komunikaci nezatížené skenování osvětlení není okem téměř zřetelný rozdíl. V případě pollingu se zapnutým skenování osvětlení již ale může dojít ke zpoždění vyššímu než 2s. V takovém případě by mohla být obsluha zmatená zda došlo k odeslání příkazu, či nikoli.

Kapitola 9

Závěr

V této práci jsem navrhl, vyrobil a úspěšně otestoval prototyp DPS pro připojení 16 podsvětlených tlačítek a vytvořil software pro obsluhu tohoto zařízení. Z důvodu nedostupnosti volně stažitelných ovladačů splňujících požadavky na komunikaci v systému, na kterém probíhalo testování, byl vytvořen zcela nový ovladač pro komunikační protokoly, zajišťující komunikaci pro 16-bit a pro 24-bit rámce, což zajišťuje možnost komunikace i pro použitý DALI modul připojený k PLC v řešeném DALI systému. Takto byla kompenzována nemožnost poskytnutého modulu komunikovat pomocí 24bitových rámců.

Navržený ovladač pro rozhraní DALI je možno využít pro přímé ovládání osvětlení bez použití jiných zařízení, nebo pro využití ve složitějších systémech s centrálním uzlem. Pro komunikaci s centrálním uzlem je možno využít odesílání eventů. Pro systémy podporující pouze DALI-1 je možno využít navrženého rozšíření pro cyklické dotazování (polling). Ovladač byl navržen tak, aby mohl být ve formě zdrojového kódu přenesen do jiných projektů, nebo na jiný typ mikrokontroléru, bez potřeby zásahu do části programu s datovým protokolem. Tohoto řešení bylo dosaženo použitím odděleného souboru, ve kterém jsou obsaženy funkce zajišťující komunikaci s periferiemi mikrokontroléru.

Hlavním cílem této práce bylo navrhnout a vytvořit zařízení, které řeší nahrazení dosavadního využití připojení průmyslových tlačítek, která jsou připojena přímo do řídicího PLC za použití rozšiřující karty DI. Nahrazení nebylo možno uskutečnit použitím dostupných součástek, jelikož se na trhu nevyskytuje podobné zařízení, splňující požadavky řešeného systému realizovaného v průmyslovém prostředí. Navržené zařízení disponuje ochranou galvanickým oddělením všech vstupů, dovoluje využití jiného zdroje napětí pro připojení podsvícení a jiného napájecího zdroje pro napájení systému s řídicí jednotkou. Bylo navrženo řešení připojení DPS ke sběrnici DALI, které bylo úspěšně otestováno v soustavě DALI zařízení.

Součástí práce byl návrh řešení způsobu komunikace, který je v souladu s platnými normami a současně řeší nestandardní situaci nemožnosti hlavního uzlu komunikace po 24bitovém rámci. Tato nemožnost vyplývala z využití DALI-1 v modulu propojujícím sběrnici DALI a řídicí jednotku

PLC. Zařízení bylo upraveno pro využití možnosti komunikace modulu, a vystupuje na sběrnici DALI jako jedno ze zařízení typu Controll gear (světlo). s tímto zařízením může být komunikováno pomocí 16-bit rámcem, nastavovat a měnit hodnoty podsvětlení, popřípadě se dotázat na event flag vystavený tlačítka.

V této práci jsem popsal různé metody, které je možné využít pro řešení komunikace. Z těchto metod jsem vybral a odůvodnil volbu použitou pro řešení na upravované soustavě. Celý program jsem řešil v jazyce C za použití objektově orientovaného přístupu k návrhu programu i programování samotnému. Program je řešen dle platné normy modulárně a je upravitelný pro použití v jiných soustavách DALI. Program ovladače byl vytvořen s výhledem pro použití s možností výměny řídicího mikrokontroléru bez zásahu do komunikačního protokolu. Tohoto bylo docíleno vytvořením odděleného souboru zajišťujícím práci s periferiemi mikrokontroléru. Takto navržená záměna byla otestována při přenosu mezi mikrokontroléry STM32 a ATMEGA328P.

Program je dále připravený pro úpravu a další možnosti rozšiřování jeho schopností. Při programování zařízení za použití navrženého ovladače je již přístupováno k odesílání a příjmu zpráv po sběrnici DALI použitím výčtového datového typu usnadňujícího další programování. Zpráva je rozdělena na několika základních parametrů, dle kterých se následně kóduje nebo dekóduje numerická zpráva.

Na vytvořeném zařízení jsem otestoval, jaké výhody plynou z využití komunikace pomocí posílání eventů oproti cyklickému dotazování (pollingu). Bylo zjištěno, že zasílání eventů je rychlejší, než polling. Pokud by neprobíhalo skenování stavu světel v celé soustavě, byl by rozdíl více než dvojnásobný, ale okem je stále téměř nepostřehnutelný. K hlavnímu rozdílu dochází při spuštění skenování osvětlení, při kterém může dojít k malému zpoždění eventu. Při použití pollingu dochází k mnohem značnějšímu zpoždění a v některých případech přesahující i 2 s, což může vést ke zmatení obsluhy a pocitu, že tlačítka nezaznamenala stisknutí.

Využitím jiného modulu, připojeného k řídicí jednotce PLC, který poskytuje možnost komunikace po 24bitových rámcích, by se prokazatelně zrychlila komunikace a zabránilo by se vytváření časů, během kterých se stisknutí tlačítka projeví až po dlouhé době. Jinou možností je upravit komunikační protokol v řídicí jednotce PLC tak, aby byla tlačítka častěji dotazována (zvýšení frekvence pollingu). Snížení periody dotazování tlačítek vede ke zkrácení času pro skenování stavu podsvětlení. Tímto dojde ke značnému prodloužení doby naskenování celé soustavy.

Práci je možno dále rozšiřovat o další ovládací prvky, pro které je již touto prací vytvořen použitelný ovladač, který poskytuje funkce pro odesílání a přijímání libovolné zprávy.

Literatura

1. ZHANG, Yuejun; ZHOU, Ping; WU, Mingguang. Research on DALI and Development of Master-Slave module. In: *2006 IEEE International Conference on Networking, Sensing and Control*. 2006, s. 1106–1110. Dostupné z DOI: 10.1109/ICNSC.2006.1673307.
2. WANG, Shun-Chung; LIU, Yi-Hua; CHEN, Yuan-Lin; CHEN, Jyun-Yan. Development of DALI-based electronic ballast with energy saving control for ultraviolet lamps. In: *2010 8th IEEE International Conference on Industrial Informatics*. 2010, s. 214–219. Dostupné z DOI: 10.1109/INDIN.2010.5549427.
3. DUNNING Scott C.; Thumann, Albert. *Efficient Lighting Applications and Case Studies*. 17.1 Overview. River Publishers, 2013. ISBN 978-0-88173-552-9. Dostupné také z: <https://app.knovel.com/hotlink/khtml/id:kt010QOMP4/efficient-lighting-applications/intelligen-overview>.
4. CSC., Ing. Jitka Machatá. *Digitální adresovatelné rozhraní pro osvětlení - Část 101: Obecné požadavky - Komponenty systému*. 2015. Tech. zpr., ČSN/EN 62386-101:2015. Úřad pro technickou normalizaci.
5. HEIN, P.F. DALI-a digital addressable lighting interface for lighting electronics. In: *Conference Record of the 2001 IEEE Industry Applications Conference. 36th IAS Annual Meeting (Cat. No.01CH37248)*. 2001, sv. 2, 901–905 vol.2. Dostupné z DOI: 10.1109/IAS.2001.955559.
6. WU, Man-Lin; KUNG, Che-Min; LIN, Yi-Nan. DALI-2 Intelligent Lighting Control System. In: *2020 International Symposium on Computer, Consumer and Control (IS3C)*. 2020, s. 158–161. Dostupné z DOI: 10.1109/IS3C50286.2020.00048.
7. CSC., Ing. Jitka Machatá. *Digitální adresovatelné rozhraní pro osvětlení - Část 102: Obecné požadavky - Ovládací zařízení*. 2015. Tech. zpr., ČSN/EN 62386-102:2015. Úřad pro technickou normalizaci.
8. CSC., Ing. Jitka Machatá. *Digitální adresovatelné rozhraní pro osvětlení - Část 103: Obecné požadavky - Řídící zařízení*. 2015. Tech. zpr., ČSN/EN 62386-103:2015. Úřad pro technickou normalizaci.

9. SINHA, Aniruddha; SHARMA, Shubham; GOSWAMI, Puja; VERMA, Vivek Kumar; MANAS, Munish. Design of an energy efficient Iot enabled smart system based on DALI network over MQTT protocol. In: *2017 3rd International Conference on Computational Intelligence Communication Technology (CICT)*. 2017, s. 1–5. Dostupné z DOI: 10.1109/CICT.2017.7977309.
10. TOOLEY, Mike. *Electronic Circuits-Fundamentals & Applications*. Routledge, 2007.
11. RITCHIE, Dennis M; KERNIGHAN, Brian W; LESK, Michael E. *The C programming language*. Prentice Hall Englewood Cliffs, 1988.
12. NORRIS, Donald. *Programming with STM32: Getting started with the Nucleo Board and C/C++*. McGraw-Hill Education, 2018.
13. BRICH, Josef et al. Realizace zařízení pro měření provozních a jízdních vlastností elektrokola. 2020.
14. KAZDEROVÁ, Jaroslava. Elektromagnetická interference a její vyhodnocování. 2010.
15. LI, Huadong; WU, Mingguang; ZHONG, Yufang. Development and research of lighting system based on DALI. In: *2008 3rd IEEE Conference on Industrial Electronics and Applications*. 2008, s. 1302–1307. Dostupné z DOI: 10.1109/ICIEA.2008.4582728.
16. JIA, Jieshu; GAO, Ruilin; XIE, Baozhuo; JIA, Dao; WEI, Feiming; ZHANG, Yuan; LIANG, Zichang; YUAN, Chengxun. Experimental Study of Scattering Characteristics on DALI Controlled Plasma-covered Target. In: *2021 IEEE 4th International Electrical and Energy Conference (CIEEC)*. 2021, s. 1–4. Dostupné z DOI: 10.1109/CIEEC50170.2021.9510669.
17. LIANG, Tsorng Juu; HUANG, Jhen Fong; YADAV, Pawan Kumar. Design and implementation of dimmable LED control circuit with DALI protocol. In: *2016 IEEE International Conference on Power and Energy (PECon)*. 2016, s. 121–126. Dostupné z DOI: 10.1109/PECON.2016.7951545.

Přílohy