

Využití mikropočítače k řízení světelných spotřebičů s využitím sběrnice DALI

Using Microcomputers for Lighting Appliance Control Using a
DALI Bus

Bc. Jan Přibyslavský



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan Přibyslavský**
Osobní číslo: **A13817**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**

Téma práce: **Využití mikropočítače k řízení světelných spotřebičů s využitím sběrnice DALI**

Téma anglicky: **Using Microcomputers for Lighting Appliance Control Using a DALI Bus**

Zásady pro vypracování:

1. Prostudujte standard DALI (Digital Addressable Lighting Interface) určený zejména pro ovládání světelných spotřebičů v domácnosti.
2. Seznamte se s normou IEC 62386. Vyberte vhodnou platformu pro realizaci jednotky master, která zároveň umožňuje konfiguraci a komunikaci s PC.
3. Navrhněte a realizujte dvě jednotky slave vícenásobný vypínač s indikačními prvky a jednotku osvětlení řízenou DALI protokolem. Pro navržené slave jednotky použijte mikrokontrolér Atmel.
4. V libovolném programovacím jazyce vytvořte PC aplikaci, která umožní celý systém konfigurovat.
5. K vytvořené aplikaci vytvořte manuál.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy – linkery, práce s ATMELE AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Vyd. 1. Praha: BEN, 2003, 279 s. ISBN 80-730-0077-6.
2. LÁNÍČEK, R. Elektronika obvodů, součástky, děje: Obvodů. Součástky. Děje. 1. vyd. Praha: BEN – technická literatura, 1998, 478 s. ISBN 80-860-5625-2.
3. SHARP, John. Microsoft Visual C# 2010: krok za krokem. Vyd. 1. Brno: Computer Press, 2010, 696 s. ISBN 978-80-251-3147-3.
4. BARNETT, Richard H, Larry O'CULL a Sarah COX. Embedded C programming and the Atmel AVR. 2nd ed. Clifton Park, NY: Thomson Delmar Learning, c2007, xxii, 532 p. ISBN 14-180-3959-4.
5. NIEMIROWSKI, Grzegorz. Serial port and microcontrollers: principles, circuits, and source codes. 1st ed. [Charleston, SC: G. NiemirowskiNew], 2013. ISBN 978-148-1908-979.
6. WOLF, R a Dick BLOCK. Scene design and stage lighting. Tenth edition. xxiii, 645 pages. ISBN 11-113-4443-4.
7. Building automation: control devices and applications. Homewood, Illinois: American Technical Publishers, Inc., c2008, 407 p. ISBN 08-269-2000-4.

Vedoucí diplomové práce:

Ing. Tomáš Sysala, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání diplomové práce:

5. února 2016

Termín odevzdání diplomové práce:

20. května 2016

Ve Zlíně dne 5. února 2016



doc. Mgr. Milan Adámek, Ph.D.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Diplomová práce se zabývá teoretickými i praktickými aspekty sběrnice DALI jakožto moderní technologie vyvinuté pro účely řízení světelných soustav. Součástí práce je teoretický úvod do problematiky, nastínění základních principů komunikace od fyzické vrstvy až po úroveň realizace složitějších datových transakcí mezi jednotlivými zařízeními. Práce se rovněž zabývá vývojem dvou typů zařízení využívajících tuto technologii a také implementací sofistikované PC aplikace, která umožňuje světelnou soustavu přes DALI sběrnici řídit. Praktické využití technologie je demonstrováno na uceleném systému tvořeným hardwarovými i softwarovými komponenty.

Klíčová slova: světelné soustavy, řízení světel, inteligentní budovy, automatizace osvětlení, DALI sběrnice

ABSTRACT

This master's thesis concerns with both theoretical and practical aspects of the DALI protocol as a modern technology developed for the purposes of controlling lighting appliances. The paper involves introduction to the topic and is outlining the basic principles from the physical layer up to the level of realization of complex transactions between individual devices. The thesis also describes a development of two different types of devices utilizing this technology as well as implementation of a sophisticated PC based application which allows a lighting appliance to be controlled over the DALI bus. The practical utilization of this technology is demonstrated using an integrated system comprising both software and hardware components.

Keywords: lighting appliances, lighting control, intelligent buildings, lighting automation, DALI bus

PODĚKOVÁNÍ

Chtěl bych poděkovat svému vedoucímu diplomové práce, Ing. Tomáši Sysalovi, Ph.D., za ochotu mojí práci vést a za cenné rady při jejím vypracování.

Dále bych rád poděkoval svému bývalému kolegovi, Ing. Ondřeji Pavelkovi, který se mi věnoval na začátku mého úsilí a osvětlil mi základy problematiky.

Rovněž bych rád poděkoval svému současnému zaměstnavateli, společnosti Baumer s.r.o., za podporu při studiu a ochotu tolerovat moji absenci v zaměstnání, kterou si tato práce vyžádala.

Jméno, příjmení: Bc. Jan Přibyslavský

Název bakalářské/diplomové práce: Využití mikropočítače k řízení světelných spotřebičů s využitím sběrnice DALI

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

.....
podpis diplomanta

Obsah

ÚVOD	9
I. TEORETICKÁ ČÁST	10
1 SBĚRNICE DALI.....	11
1.1 STRUČNÝ POPIS ROZHRANÍ	11
1.2 PRINCIP KOMUNIKACE	11
1.3 TOPOLOGIE.....	15
1.4 BEZPEČNOSTNÍ ASPEKTY	16
1.5 SROVNÁNÍ S PODOBNÝMI TECHNOLOGIEMI.....	17
1.6 MOTIVACE K POUŽÍVÁNÍ.....	18
2 MOŽNOSTI PRAKTICKÝCH APLIKACÍ	21
2.1 HOTOVÁ ŘEŠENÍ	21
2.2 VÝVOJ VLASTNÍCH ZAŘÍZENÍ.....	23
II. PRAKTICKÁ ČÁST	26
3 PRAKTICKÁ REALIZACE DALI SLAVE JEDNOTKY.....	27
3.1 VOLBA VÝCHOZÍHO KONCEPTU	27
3.2 KONCEPT DALI SLAVE JEDNOTKY VSTUPNĚ-VÝSTUPNÍ	29
3.3 KONCEPT DALI SLAVE JEDNOTKY SENZOROVÉ	30
3.4 NÁVRH ZAPOJENÍ.....	31
3.4.1 FYZICKÁ VRSTVA DALI	31
3.4.2 NAPÁJECÍ ČÁST	32
3.4.3 MIKROKONTROLÉR	33
3.4.4 VSTUPY	34
3.4.5 VÝSTUP	35
3.4.6 MĚŘENÍ HODNOTY OKOLNÍHO OSVĚTLENÍ.....	37
3.5 NÁVRH A REALIZACE PLOŠNÉHO SPOJE	40
3.5.1 TECHNOLOGICKÉ POŽADAVKY.....	40
3.5.2 NÁVRH	41
3.5.3 OSAZENÍ.....	43
3.6 VÝVOJ FIRMWARE	46

3.6.1	VÝCHOZÍ KONCEPT	46
3.6.2	ARCHITEKTURA	47
3.6.3	ZPRACOVÁNÍ SIGNÁLŮ Z FYZICKÉ VRSTVY (DRIVERS).....	47
3.6.4	ZPRACOVÁNÍ RÁMCŮ (DALI SERVICES).....	49
3.6.5	APLIKAČNÍ VRSTVA (DALI APPLICATION)	50
3.6.6	HLAVNÍ SMYČKA PROGRAMU	58
4	PRAKTICKÁ REALIZACE TESTOVACÍ FIXTURY	59
4.1	ÚČEL TESTOVACÍ FIXTURY	59
4.2	POPIS TESTOVACÍ FIXTURY	59
4.3	VOLBA JEDNOTKY DALI-MASTER	61
4.4	ZAPOJENÍ TESTOVACÍ FIXTURY	62
4.5	KONFIGURACE JEDNOTLIVÝCH MODULŮ	64
5	ŘÍDICÍ A KONFIGURAČNÍ PC-APLIKACE	65
5.1	POPIS PC-APLIKACE	65
5.2	VÝCHOZÍ KONCEPT	65
5.3	REALIZACE PC-APLIKACE.....	66
5.3.1	POUŽITÉ TECHNOLOGIE	66
5.3.2	ARCHITEKTURA	66
5.3.3	GRAFICKÉ ROZHRANÍ	69
5.3.4	SPRÁVA KONFIGURAČNÍCH DAT	70
5.3.5	ŘÍDICÍ FUNKCE.....	74
5.3.6	KOMUNIKACE S JEDNOTKOU DALI-MASTER	79
5.3.7	PŘÍMÝ PŘÍSTUP NA DALI SBĚRNICI	80
5.3.8	MONITORING	81
	ZÁVĚR.....	82
	CONCLUSION	83
	SEZNAM POUŽITÉ LITERATURY	84
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	86
	SEZNAM PŘÍLOH	87

ÚVOD

Asi by bylo zcela zbytečné uvádět skutečnost, že elektrické osvětlení je dnes běžnou výbavou většiny obyvatelných budov. Je však třeba si uvědomit, že klasický koncept vypínač-žárovka zůstal v mnoha případech prakticky nezměněn od konce 19. století, kdy se začal hojněji rozšiřovat. Je to docela zajímavé, protože ostatní technologie založené na elektrotechnice či informatice prošly za tu dobu ohromným vývojem. Přesto však při ohlédnutí za poslední dvě dekády můžeme pozorovat určitý rozvoj celé řady technologií zaměřených právě na řízení osvětlení.

Stále častěji se dnes setkáváme s pojmem inteligentní budovy. Jedná se o stavby, které jejich obyvatelům zajišťují optimální vnitřní prostředí, jsou ekonomicky i energeticky efektivní a mnohdy nabízejí víceúčelové využití. To vše za použití moderní techniky, zejména z oblastí informačních technologií a automatizace a řízení, nevyjímaje sofistikovaných způsobů realizace osvětlení.

Ačkoliv se tato diplomová práce nezabývá přímo problematikou inteligentních budov, poskytuje alespoň malé okénko do tohoto světa přiblížením právě diskutovaného řízení světelných systémů.

Tato diplomová práce je určena každému, koho daná problematika zajímá, ale především každému, kdo zamýšlí v dané oblasti uskutečňovat jakýkoliv vývoj.

I. TEORETICKÁ ČÁST

1 SBĚRNICE DALI

1.1 Stručný popis rozhraní

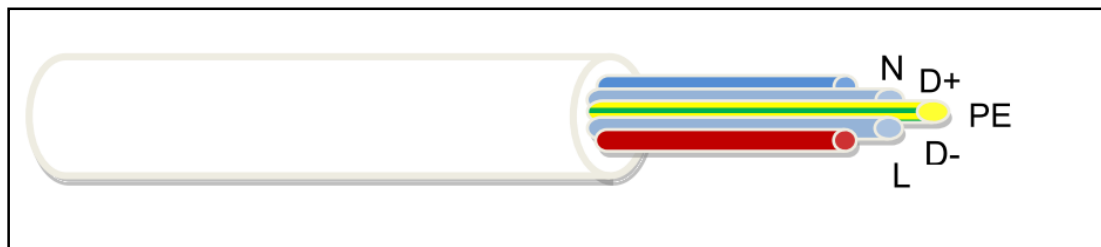
Sběrnice DALI (z angl. Digital Addressable Lighting Interface), vycházející z mezinárodní technické normy IEC 62386, je digitální sběrnice určená primárně pro řízení světelných soustav. Jedná se o otevřený protokol tzn. umožňující mnoha různým výrobcům, aby jej využívali. DALI nabízí jednotné rozhraní pro všechny účastníky na sběrnici. Z hlediska funkce jednotlivých zařízení můžeme mluvit o jednotkách **master** a **slave**.

Zařízením master zpravidla rozumíme řídicí jednotku, která zahajuje komunikaci, oslovuje jednotlivá zařízení slave a musí být na sběrnici vždy alespoň jedna. Zařízení slave je potom vzdálená jednotka, která plní funkci na základě příkazu z jednotky master. Může, ale nemusí, vysílat data zpět, ale vždy jen na základě nějakého předchozího dotazu.

1.2 Princip komunikace

Fyzická vrstva

Pro přenos dat na sběrnici DALI slouží dva vodiče, na které se všechna zařízení připojují paralelně. Jedna samostatná větev umožňuje připojit až 64 jednotek v libovolné kombinaci slave nebo master. Rychlost komunikace je pevně stanovena na 1200 Bd. Protokol DALI byl vyvinut s myšlenkou, že pro datové vodiče budou použity stejné typy jako pro rozvod 230 VAC a veškeré tyto vodiče budou součástí jednoho kabelu, viz následující obrázek.

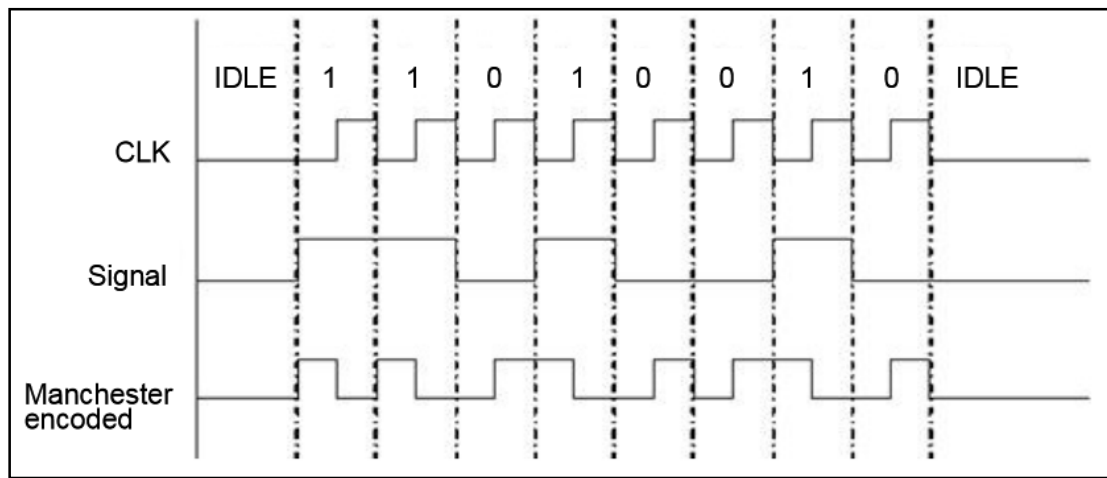


Obrázek 1 - Rozvod signálů sběrnice DALI (D+ a D-) společně se silnoproudým rozvodem (N, L, PE), (1)

Kódování Manchester

Přenos dat na sběrnici využívá kódování Manchester, kde z hlediska informace není důležitá úroveň signálu, ale přechod mezi úrovněmi. Náběžná hrana znamená logickou 0, sestupná logickou 1. Napětíové úrovně signálu jsou typicky 0-16 V (2).

Na následujícím obrázku je podrobněji znázorněno kódování jednotlivých bitů.

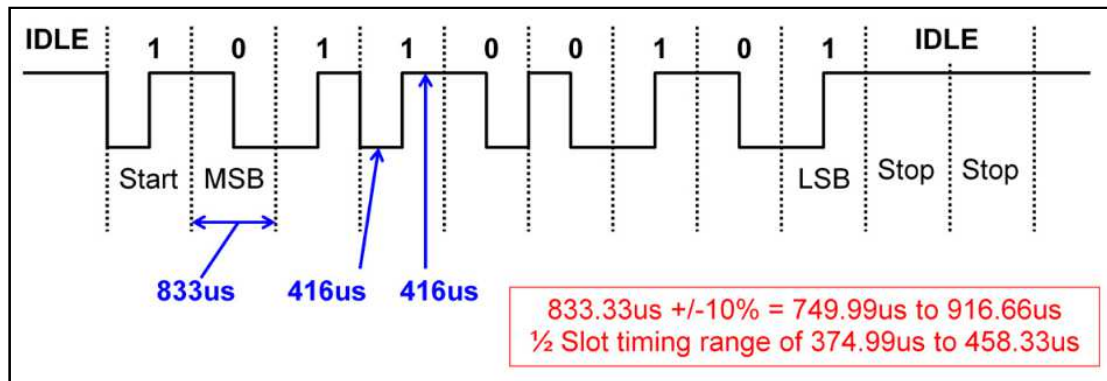


Obrázek 2 - Ukázka kódování Manchester (2).

Rámce

Při přenosu dat jsou na sběrnici DALI z jednotlivých bitů skládány tzv. rámce. Rámec může být jak řídicí (forward-frame), kterým nejčastěji jednotka master oslovuje jednotky slave, tak i rámec zpětný (backward-frame), kterým tyto slave jednotky odpovídají.

Rámec je vždy zahájen jedním start bitem a ukončen dvěma stop bity. Délka jednoho bitu je vždy stejná a při rychlosti přenosu dat 1200 Bd odpovídá času 833,33 us. Přípustná tolerance v délce trvání je +/- 10%.



Obrázek 3 - Časování jednoho rámce na sběrnici DALI (2).

Kromě start a stop bitů obsahují řídicí rámce (forward-frames) navíc ještě dalších 8 bitů adresových a 8 bitů datových. Zpětné rámce (backward-frames) obsahují navíc pouze 8 bitů datových.



Obrázek 4 - Řídicí rámec sběrnice DALI (2).



Obrázek 5 - Zpětný rámec sběrnice DALI (2).

Na předchozích dvou obrázcích jsou naznačeny oba typy rámců na sběrnici DALI. Významy jednotlivých bitů jsou následující:

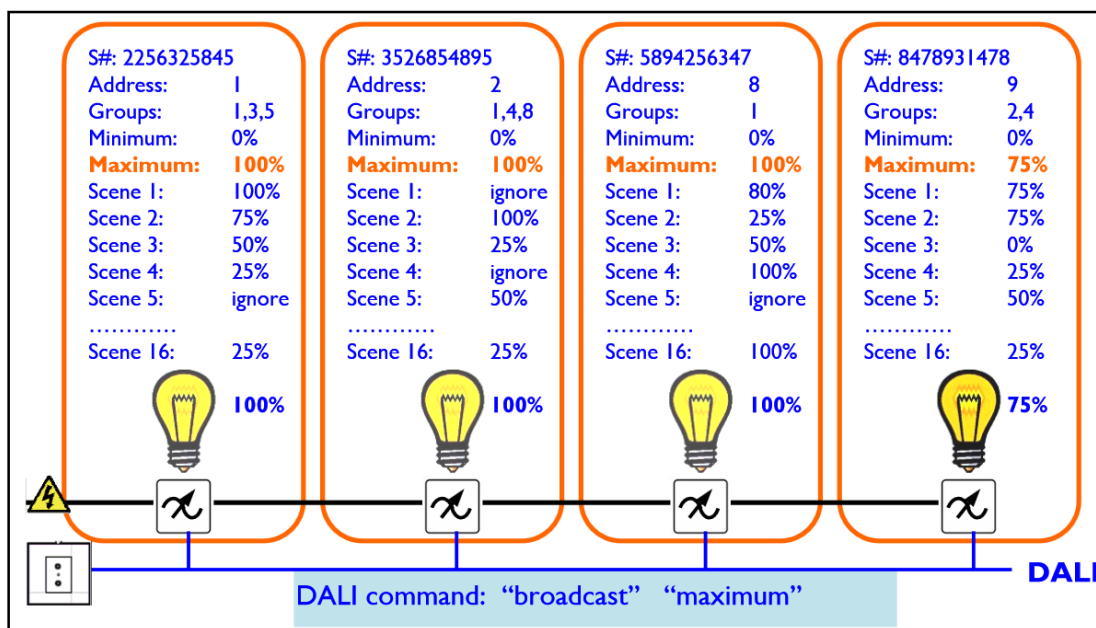
- s - start bit
- YAAA AAAS - adresové bity
- XXXX XXXX - datové bity
- I - stop bit

Adresy, skupiny a příkazy

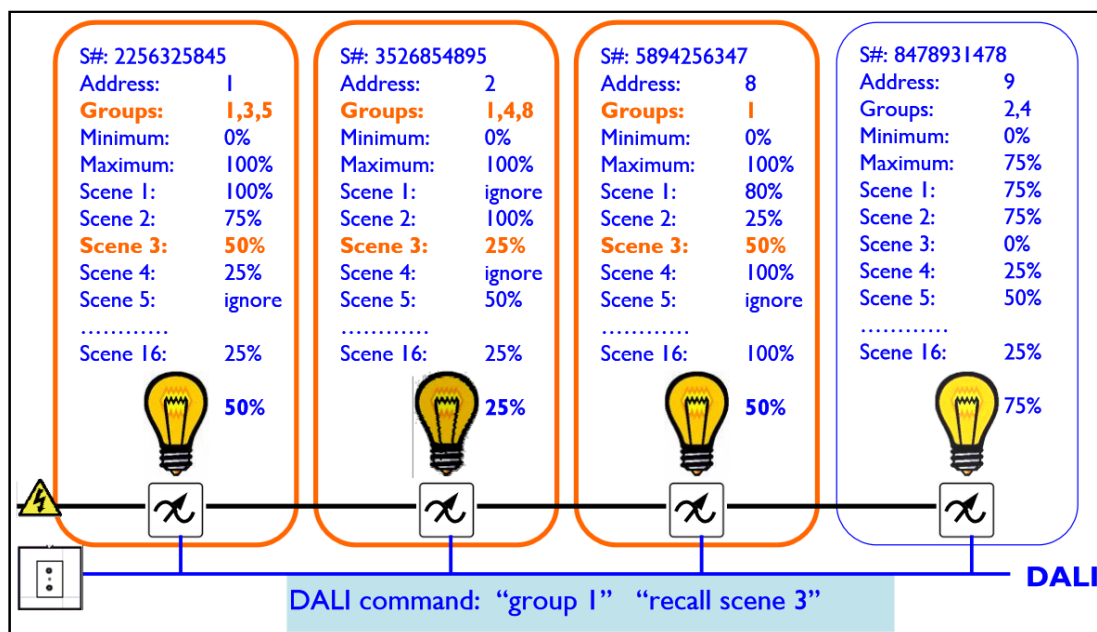
Každé slave zařízení na sběrnici DALI má svoji adresu. Je-li ze zařízení master vyslán řídicí rámec a jeho adresová část obsahuje adresu právě tohoto slave zařízení, tak na ni toto zařízení zareaguje. Reakce je zpravidla taková, že si slave zařízení přečte i datovou část přijatého rámce, kde může být např. příkaz, který má toto zařízení vykonat. Jelikož je datová část rámce 8 bitů, může zařízení tímto způsobem vykonat jeden z 256 možných příkazů.

Kromě samotných adres může být každé zařízení přiřazeno do jedné nebo více skupin. Maximální počet celkových skupin je 16. Vyšle-li master zařízení řídicí rámec, kde význam adresové části je voláním skupiny, zareaguje na tento rámec každé zařízení, které je do volané skupiny přiřazeno. Takto je možné jedním voláním obsloužit několik zařízení aniž by bylo nutné obsluhovat každé z nich zvlášť postupným voláním jejich individuálních adres.

Na následujících obrázcích jsou uvedeny příklady, jak by mohla komunikace pomocí adres, skupin a příkazů probíhat.



Obrázek 6 - "Broadcast" příkaz, který způsobí rozsvícení všech světel na maximum (3).

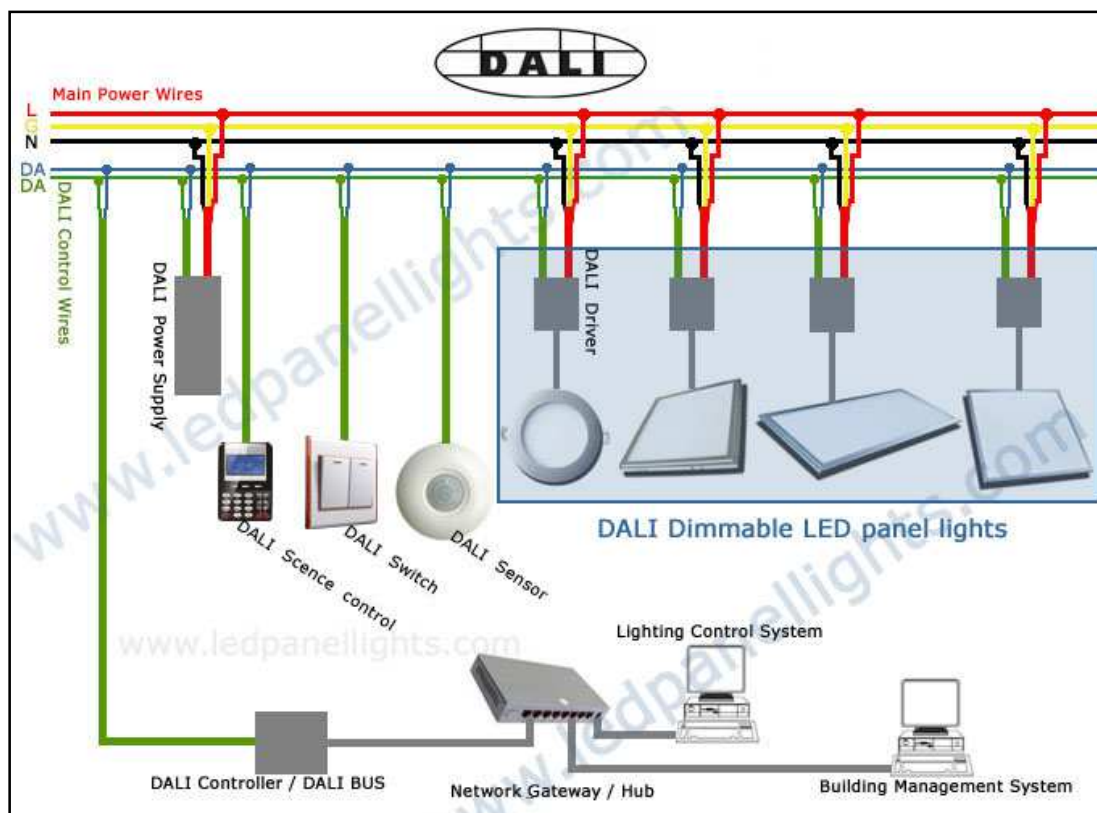


Obrázek 7 - Příkaz skupině 1, který způsobí přechod do scény 3 (3).

1.3 Topologie

Topologie sběrnice DALI zpravidla zahrnuje paralelní propojení všech účastníků na jedné síti. Pojmem síť zde rozumíme uskupení maximálně 64 jednotek, kde je vždy obsažen napájecí zdroj sběrnice DALI, řídicí jednotka (DALI controller) a alespoň jedno zařízení sloužící jako zdroj světla. Řídicí jednotka může být dále připojena přes jinou sběrnici k dalšímu, rozsáhlejšímu systému. Potřebujeme-li na sběrnici DALI ovládat více, než 64 jednotek, je nutné sběrnici rozšířit na další samostatnou síť pomocí tzv. brány (gateway) případně může být více DALI sítí sjednocených pomocí jejich řídicích jednotek, které spolu vzájemně komunikují jiným způsobem.

Na obrázku níže je vyobrazena topologie jedné samostatné DALI sítě.



Obrázek 8 - Znázornění topologie sběrnice DALI (4).

1.4 Bezpečnostní aspekty

Bezpečnost z hlediska elektrotechniky

Vzhledem k tomu, že je možné provozovat datové vodiče DALI sběrnice společně s rozvodem 230VAC a to i v jednom kabelu, není možné řadit samotnou DALI sběrnici do kategorie SELV (z angl. Separated Extra Low Voltage) z čehož plyne i nutnost galvanického oddělení jednotlivých zařízení. Každé zařízení by navíc mělo být schopno na svých DALI svorkách bezpečně vydržet síťové napětí 230VAC v případě poruchy izolace mezi vodiči.

Ze stejného důvodu je rovněž kladen požadavek na oba datové kabely, kde každý musí splnit stejná kritéria, jako kabel pro síťové napájení. Minimální průřez kabelu 1 mm^2 , izolace do 600V a úbytek napětí maximálně 2V na celé délce rozvodu (max 300m) (5).

Specifikace DALI rovněž definuje požadavek na bezpečný spolehlivý chod všech zařízení nezávisle na polaritě jejich připojení na sběrnici. Tato funkce je realizovaná na fyzické vrstvě každého zařízení. **Veškerá zařízení, která byla prakticky vyvíjena a provozována v rámci této práce nepracovala s napětím vyšším než 24 V.**

Bezpečnost a integrita dat

Jelikož DALI využívá Manchester kódování a poměrně velký rozsah napěťových úrovní s relativně malou přenosovou rychlostí, dá se hovořit o sběrnici, která je odolná vůči šumu a jinému rušení, které zcela jistě díky přítomnosti síťových napájecích kabelů může být přítomno. Z hlediska bezpečnosti a integrity dat však není definována žádná metoda, která by byla na sběrnici využita (např. kontrola parity nebo CRC) a tak je přece jen nutné počítat s chybami v přenosu.

1.5 Srovnání s podobnými technologiemi

Sběrnice DALI je poměrně nová technologie a její vývoj prošel určitými fázemi. Za zmínku jistě stojí její stále často používaný předchůdce - technologie DSI (6) a rovněž technologie, která pro řízení jasu světel využívá analogové napětí 0-10V (7).

DSI byla vyvinuta rakouskou firmou Tridonic (8) v roce 1991 a stala se první digitální technologií používanou pro ovládání světelných soustav. Stejně jako DALI je postavena na kódování Manchester a přenosové rychlosti 1200 Bd. Jedná se však o protokol, který není možné používat ve sběrnicovém zapojení a kde každé slave zařízení má svoje vlastní signálové vodiče k jednotce master. Toto by mohlo být považováno i za výhodu - odpadá totiž nutnost adresování jednotlivých zařízení a v případě poruchy některého z nich lze toto pouze odpojit a nahradit novým. Velkou nevýhodou takového řešení je však značné množství kabelů navíc a nemožnost sdružování více zařízení do skupin.

Půjdeme-li více do minulosti, setkáme se s dnes ještě stále využívanou technologií 0-10V. Jedná se o způsob, jak dálkově řídit výkon světelných soustav pomocí analogového signálu. Zařízení navržená pro tuto technologii dokáží na základě napěťové úrovně řídicího

signálu modulovat výkonový výstup do světelného zdroje. Může se jednat o lineární závislost, kdy rozsahu 0-10V odpovídá 0-100% světelného výkonu, avšak křivky závislosti mohou být i jiné. Stejně jako v případě technologie DSI, je i zde nutné vést ke každému zařízení samostatný pár vodičů. Vzhledem k použití analogového signálu, který je velmi náchylný na rušení, je obtížné vést tyto kabely společně se síťovým napájením 230VAC. Rovněž je nutné kompenzovat úbytky napětí na vedení a vzdálenější zařízení patřičným způsobem kalibrovat. Určitou výhodou tohoto analogového řešení však může být skutečnost, že zařízení lze realizovat plně v analogovém režimu a není třeba při jeho konstrukci využívat žádný mikrokontrolér.

1.6 Motivace k používání

Adresovatelnost

Z již uvedených informací v této práci může vyplynout jedna z hlavních výhod sběrnice DALI - adresovatelnost jednotlivých zařízení.

Uvažujme síť světelných zdrojů a ovládacích prvků (např. vypínačů nebo stmívačů), které jsou určitým způsobem fyzicky rozmístěny v budově. V případě běžné instalace jsou tyto prvky pevně propojeny a jakákoliv rekonfigurace celého systému je bez fyzických zásahů do samotné instalace jen velmi obtížná. V případě adresovatelnosti jednotlivých zařízení však každé z nich vystupuje na celé síti jako samostatná jednotka a je jen na řídicím systému (jednotce master) jakým způsobem je k sobě přiřadí. Změna konfigurace systému je pak jen otázkou změny přiřazení v software. Rovněž se zde nabízí možnost přiřadit více kontrolních prvků k jednomu světelnému zdroji (např. osvětlení chodeb a schodišť), případně více světelných zdrojů k jednomu kontrolnímu prvku (osvětlování velkých kanceláří a hal, vytváření světelných scén na jevišti či v posluchárnách atd...).

Automatizace osvětlení

Jelikož jsou všechna zařízení na sběrnici DALI řízena programovatelným systémem, je velmi snadné docílit samoregulace osvětlení v daných prostorách a v daném čase. Můžeme například uvažovat halu, která slouží během pracovní doby k setkávání většího počtu lidí a v daných časových intervalech jsou zde zvýšené požadavky na osvětlení. Mimo tyto doby je však prostor pouze průchozí a na jeho osvětlení stačí mnohem menší intenzita. Toto je

možné dosáhnout časovým naprogramováním systému. Má-li tato místnost přístup k dennímu světlu, je samozřejmě možné na základě čidel zajistit, že požadovaná míra osvětlení je v daný okamžik vždy konstantní, nezávisle na množství světla přicházejícího z venku.

Simulace přítomnosti osob

V návaznosti na možnost automatizace osvětlovacího systému je možné sofistikovaným způsobem simulovat přítomnost osob v budovách v době, kdy nejsou obydleny. Toto je užitečné zejména pro účely ochrany majetku.

Otevřenost protokolu

Na rozdíl od technologie DSI, která byla po dlouhou dobu omezená pouze na výrobky společnosti Tridonic, je sběrnice DALI zcela otevřená tudíž podporovaná celou řadou nejrozličnějších výrobců. Při návrhu a realizaci světelných soustav je tedy možné vybírat z mnohem širší nabídky produktů. Nabízejí se i možnosti vývoje vlastních zařízení.

Snadná kabeláž

Vzhledem k tomu, že datové kabely sběrnice DALI je možné rozvádět souběžně s napájením, a že každý světelný zdroj musí mít alespoň toto napájení přivedeno, je složitost kabeláže pouze taková, jaká by byla za normálních okolností při zapojení pouze světel samotných. Není třeba k žádnému zařízení přivádět jakékoliv kabely navíc (kabel pouze obsahuje o 2 vodiče více).

Možnosti přednastavených scén

Představme si konferenční místnost, která slouží k celé řadě nejrozličnějších účelů. Můžeme požadovat jiné osvětlení pro ranní business meeting, dopolední setkání u kávy, prezentaci s data-projektorem nebo relaxační aktivity. Berme v potaz, že nasvětlení místnosti nemusí být dáno pouze intenzitou celkového světla, ale i jeho barvou a případně podílem jednotlivých světelných zdrojů. Pro každou danou situaci může být v systému uložená

přesná konfigurace a tu lze pak vyvolat jednoduše stiskem tlačítka na ovládacím panelu v místnosti, případně na dálku jiným způsobem.

Obousměrná komunikace

DALI sběrnice umožňuje obousměrnou datovou komunikaci, čímž výrazně rozšiřuje možnosti použití nad rámec pouhého nastavování intenzity osvětlení. Na sběrnici je možné připojovat již zmiňovaná čidla osvětlení, ale i např. detektory pohybu, která mohou posloužit jako zdroje událostí pro další akce, či pro účely sledování využití daných prostor.

Obousměrná komunikace dává možnost jednotlivým světelným zdrojům poskytovat údaje o svojí spotřebě elektrické energie během dne, což může hrát klíčovou roli v případě nutnosti optimalizace celého systému osvětlení v budově. Jednotlivá zařízení rovněž mohou disponovat schopností diagnostiky závad a ty pak hlásit na sběrnici. V rozsáhlých kancelářích, či výrobních halách se stovkami zařízení může tato schopnost výrazně snížit náklady na údržbu.

2 MOŽNOSTI PRAKTICKÝCH APLIKACÍ

2.1 Hotová řešení

Nabídka zařízení využívajících sběrnici DALI je v dnešní době poměrně uspokojivá. Významnými výrobci v této oblasti jsou společnost Phillips (9), OSRAM (10), Helvar (11) a Tridonic (8).

Co do počtu zařízení, rozmanitá nabídka je zejména v kategorii tzv. stmívacích předřadníků (angl. dimmable ballast), což jsou elektronické prvky určené k buzení a modulaci výkonu do zářivek a nahrazují tak klasické řešení s tlumivkou a startérem. V poslední době se však obdobná zařízení využívají k modulaci výkonu pro celou řadu jiných světelných zdrojů i včetně nízkonapěťových LED diod.

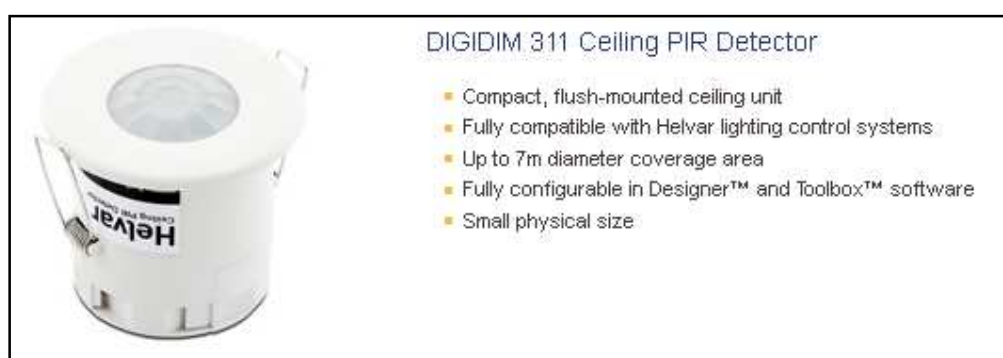
Příklady výrobků společnosti Helvar:



Obrázek 9 - DALI router, který podporuje až dvě DALI podsítě včetně jejich napájení a umožní připojení k nadřazené síti Ethernet (12).



Obrázek 10 - Ovládací prvky pro sběrnici DALI (13).



Obrázek 11 - Stropní PIR detektor umožňující reagovat na přítomnost osob v místnosti (14).



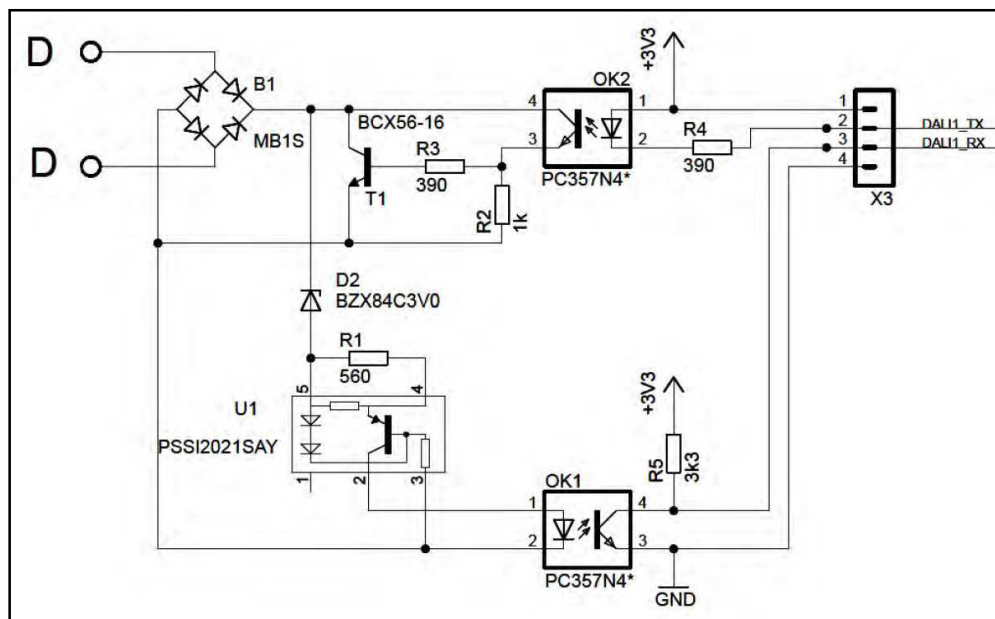
Obrázek 12 - Senzor úrovně venkovního osvětlení (14).

2.2 Vývoj vlastních zařízení

Vývojář, který by chtěl vytvářet svá vlastní zařízení pro sběrnici DALI, bude pravděpodobně muset překonat několik úskalí, jak z hlediska realizace fyzické vrstvy, tak z pohledu zpracovávání paketů v kódování Manchester.

Vzhledem k tomu, že všechna jednotlivá zařízení na sběrnici DALI jsou galvanicky oddělená, je třeba s tímto faktem počítat při návrhu zapojení. Rovněž je nutné zajistit, aby signálové vodiče DALI sběrnice nebyly závislé na polaritě zapojení. Pro realizaci výše uvedeného je možné vyjít z již některých hotových řešení. Jako demonstrativní příklad je uvedena deska od společnosti NXP **DALI slave board OM13026**.

Schéma fyzické vrstvy je znázorněno na následujícím obrázku.



Obrázek 13 - Schéma fyzické vrstvy sběrnice DALI na desce OM13026 (1).

Z obrázku je patrné, že pro galvanické oddělení byly využity optočleny (OK1 a OK2), kde každý datový směr využívá právě jeden. Jedná se o levné a efektivní řešení, které zároveň nabízí úpravu signálu na požadovaný rozsah napětíových úrovní. V tomto případě 0 - 3,3V.

Nezávislost na polaritě signálových vodičů sběrnice DALI (svorky označené písmenem D) je zajištěna usměrňovacím můstkem B1. Objeví-li se na svorkách D napětí vyšší, než je zenerovo napětí diody D2, začne protékat proud LED diodou v optočlenu OK1 a jeho tranzistor se otevře, čímž na signálu DALI1_RX zajistí logickou 0 (zenerovo napětí je v tomto případě 3V, je však třeba počítat i s otevíracím napětím diody v optočlenu, které musí být k tomu zenerovu přičteno). V opačném případě je na tomto signálu logická 1, která je zajištěná zdvihacím rezistorem R5. V případě vysílání dat (signál DALI1_TX) je logickou 0 na tomto signálu buzena dioda v optočlenu OK2 a následné otevření tranzistoru T1 způsobí zkrat mezi svorkami D. V opačném případě jsou tyto svorky ve stavu vysoké impedance.

Je zcela zřejmé, že pro realizaci vlastní slave nebo master jednotky bude nezbytné využít některý z moderních mikrokontrolérů. Signály DALI1_RX a DALI1_TX na výše uvedeném schématu mohou být k takovému mikrokontroléru přímo připojeny. Problémem zde může být použité kódování Manchester, které ne každý mikrokontrolér dokáže přímo zpracovávat. Obvod LPC1343, který je na desce OM13026 použit, umí komunikovat pouze ve standardním UART módu a pro práci s kódováním Manchester bylo nutné přijít s vlastním řešením. Vysílání dat na signálu DALI1_TX je řešeno softwarově bez využití jakékoliv sofistikované hardwarové periferie. Signál DALI1_RX, na kterém je realizován příjem dat, je zaveden na vstup časovače obvodu LPC1343, který je nastaven tak, aby periodicky vzorkoval úroveň tohoto signálu. Vzorky jsou poté vyhodnocovány pomocí softwaru.

Řešení sériové komunikace čistě pomocí softwaru může být výhodné v případě, že daný mikrokontrolér neposkytuje žádnou hardwarovou podporu pro daný sériový protokol. Je tedy možné tuto metodu aplikovat na každém mikrokontroléru s ohledem na velikost jeho paměti a výpočetní výkon.

Značnou nevýhodou tohoto řešení je však skutečnost, že vyhodnocování a zpracování každého bitu na sběrnici konzumuje část výpočetního výkonu. Rovněž software, jenž tuto funkci realizuje, zabírá místo v paměti, což ne vždy může být žádoucí. Mnohem

sofistikovanější řešení by bylo použít mikrokontrolér, jenž disponuje hardwarovou podporou pro zpracování dat kódovaných metodou Manchester.

Po krátkém průzkumu trhu můžeme objevit docela uspokojivou nabídku mikrokontrolérů, jenž požadovanou funkci nabízejí. Jako velmi vhodný kandidát by se mohl jevit kterýkoliv mikrokontrolér od společnosti Atmel (15) z řady **AT90PWM** (16). Tato řada byla speciálně navržena pro světelné aplikace a každý z jejích mikrokontrolérů nabízí rozšířenou funkci modulu UART pro 16/17 bitovou komunikaci na sběrnici DALI. Ačkoli se jedná o 8bitový mikrokontrolér, který oproti 32bitovému LPC1343 nenabízí takový výpočetní výkon, možnost zpracovávání DALI komunikace přímo na úrovni hardwaru tento nedostatek dostatečně kompenzuje. Cena obvodů z řady AT90PWM je srovnatelná s obvody řady LPC13.

II. PRAKTICKÁ ČÁST

3 PRAKTICKÁ REALIZACE DALI SLAVE JEDNOTKY

Jak už bylo zmíněno v předchozí kapitole, při realizaci vlastní DALI Slave jednotky se nabízí možnost vyjít již z některých hotových řešení, která jsou volně k dispozici. V rámci této diplomové práce taková snaha byla, avšak i přesto bylo nutné rozhodnout nad mnoha dalšími kritérii, aby bylo zřejmé, kterou konkrétní variantu zvolit.

3.1 Volba výchozího konceptu

Pro výběr vhodného výchozího konceptu byla zvolena tzv. **Pughova metoda** (17). Účelem Pughovy metody je rozdělit jedno větší rozhodnutí na několik menších, která mohou být vyhodnocena samostatně a ve výsledku pak nabídnou výsledné skóre pro každý koncept.

Několik konceptů, které se ze začátku nabízelo, bylo shrnuto do přehledné tabulky. Pro všechny tyto koncepty bylo zvoleno několik atributů, které byly považovány za důležité. U každého z těchto atributů pak bylo rozhodnuto, zdali je u daného konceptu lepší, horší nebo stejně dobrý, než u jednoho zvoleného referenčního konceptu.

Výchozí koncepty byly následující:

1. **Koncept 1** – využití hotového komerčního řešení. Jako jedna z variant by se mohla jevit koupě již hotové DALI Slave jednotky, která disponuje požadovanými vstupy a výstupy.
2. **Koncept 2** – Využití již hotové DALI Slave board OM13026. Tento modul by byl modifikován z hlediska HW i SW tak, aby vyhověl zadání diplomové práce.
3. **Koncept 3** – Využití fyzické vrstvy z OM13026 a vývoj vlastní platformy za použití mikrokontroléru Atmel podporujícího kódování Manchester.
4. **Koncept 4** – Využití fyzické vrstvy z OM13026 a vývoj vlastní platformy za použití referenčního návrhu (18), který pro své mikrokontroléry poskytuje společnost Atmel.

Atributy hodnocené pro jednotlivé koncepty byly následující:

- Cena realizace
- Dostupnost komponent
- Časová náročnost na realizaci
- Podpora výrobce komponent z hlediska uživatelského vývoje SW
- Dostupnost vývojového prostředí
- Možnosti HW modifikací
- Možnosti SW modifikací
- Vyhovění zadání diplomové práce

Pughova metoda výběru konceptu byla zpracována v následující tabulce. Koncept 1 byl zvolen jako referenční.

	Koncept			
Atribut	1	2	3	4
Cena	S	++	+	+
Dostupnost	S	+	+	+
Čas. náročnost	S	-	--	--
Podpora výrobce	S	S	-	+
Vývojové prostředí	S	S	+	+
HW možnosti modifikací	S	+	++	++
SW možnosti modifikací	S	+	+	++
Vyhovění zadání	S	+	++	++
Počet +	0	6	8	10
Počet -	0	1	3	2
Celkové skóre	0	5	5	8

Tabulka 1 – Pughova metoda výběru konceptu

Jak je z tabulky patrné, jako nejvhodnější výchozí koncept pro realizaci DALI Slave jednotky se jevil Koncept 4, tedy převzetí fyzické vrstvy z desky OM13026 a vývoj vlastní platformy za využití referenčního návrhu od společnosti Atmel. Použitým mikrokontrolérem je v tomto případě ATMEGA88. Vývojovým prostředím potom Atmel Studio 7.0.

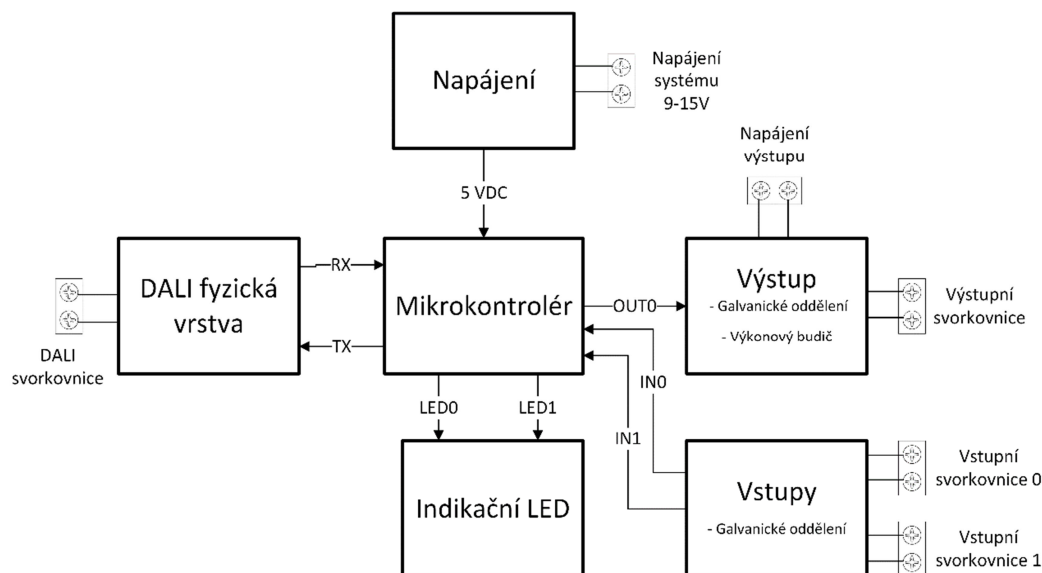
3.2 Koncept DALI Slave jednotky vstupně-výstupní

Vstupně-výstupní DALI Slave jednotky byla navržena tak, aby po všech stránkách vyhovovala zadání práce. Počet jednotek je stanoven na dva kusy, přičemž jedna musí nabízet funkci řízení světelného zdroje po sběrnici DALI a druhá vícenásobný vypínač s indikačními prvky. Pro účely zjednodušení návrhu byly tyto funkce sjednoceny a všechny je nabízí jen jedna DALI Slave jednotka. Tato je pak vyrobena dvakrát a je již na volbě uživatele, kterou se rozhodne využít jako světelný zdroj a kterou pro účel vícenásobného vypínače.

Jak už bylo uvedeno, fyzická vrstva DALI Slave jednotky byla převzata z desky OM13026. Zbývá část zapojení poté vychází z referenčního návrhu od společnosti Atmel.

Rozhraní DALI, stejně tak všechny vstupy a výstupy, byly realizovány s ohledem na galvanické oddělení pomocí optočlenu.

Blokové schéma vstupně-výstupní jednotky je znázorněno na následujícím obrázku:



Obrázek 14 – Blokové schéma DALI Slave jednotky vstupně-výstupní

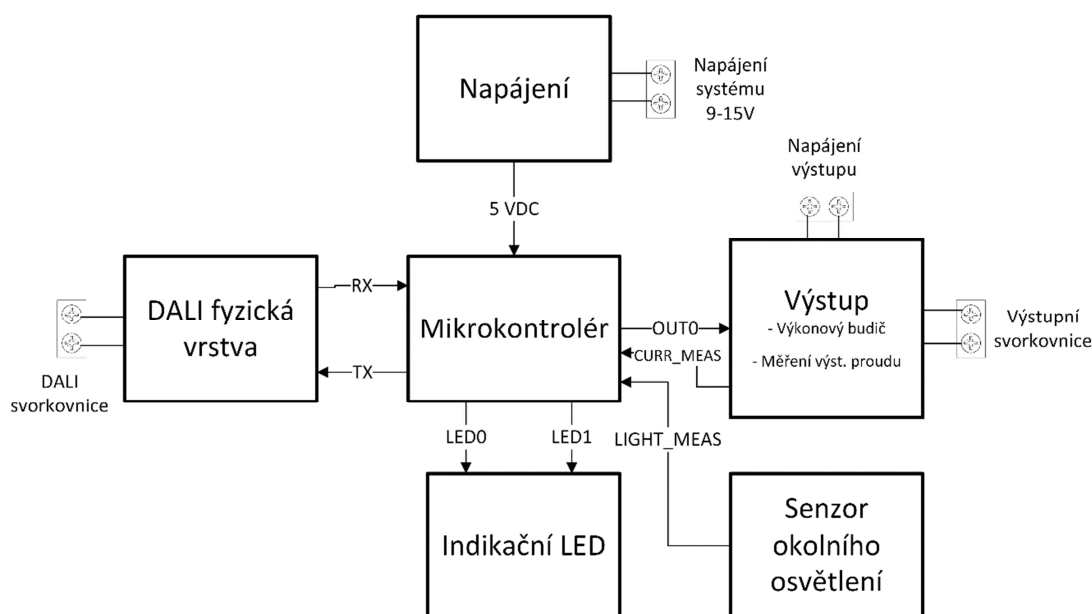
3.3 Koncept DALI Slave jednotky senzorové

Senzorová DALI Slave jednotka byla navržena až dodatečně za účelem demonstrace pokročilejších možností sběrnice DALI. Po stránce hardwarové i softwarové vychází senzorová jednotka z velké části ze vstupně-výstupní. Oproti té však přinesla tyto změny:

- Odebrání uživatelských vstupů (tlačítka)
- Odebrání galvanického oddělení výstupu
- Přidání rezistoru pro účely měření proudu do zátěže na výstupu
- Přidání napěťového děliče s fotorezistorem pro účely měření intenzity okolního osvětlení.

Kromě možností měření intenzity okolního osvětlení a detekci chyby v zátěži (v případě, že protékající proud není takový, jaký je očekáván) nabízí přidání třetí Slave jednotky do systému ještě jednu možnost – použití DALI skupin. V případě, že bychom se rozhodli provozovat pouze dvě Slave jednotky, každou v jiné skupině, nebyl z praktického hlediska žádný rozdíl mezi adresováním pomocí přímých adres nebo skupin.

Stejně jako pro vstupně-výstupní jednotku, vznikl i pro jednotku senzorovou koncept v podobě blokového schématu:



Obrázek 15 – Blokové schéma DALI Slave jednotky senzorové

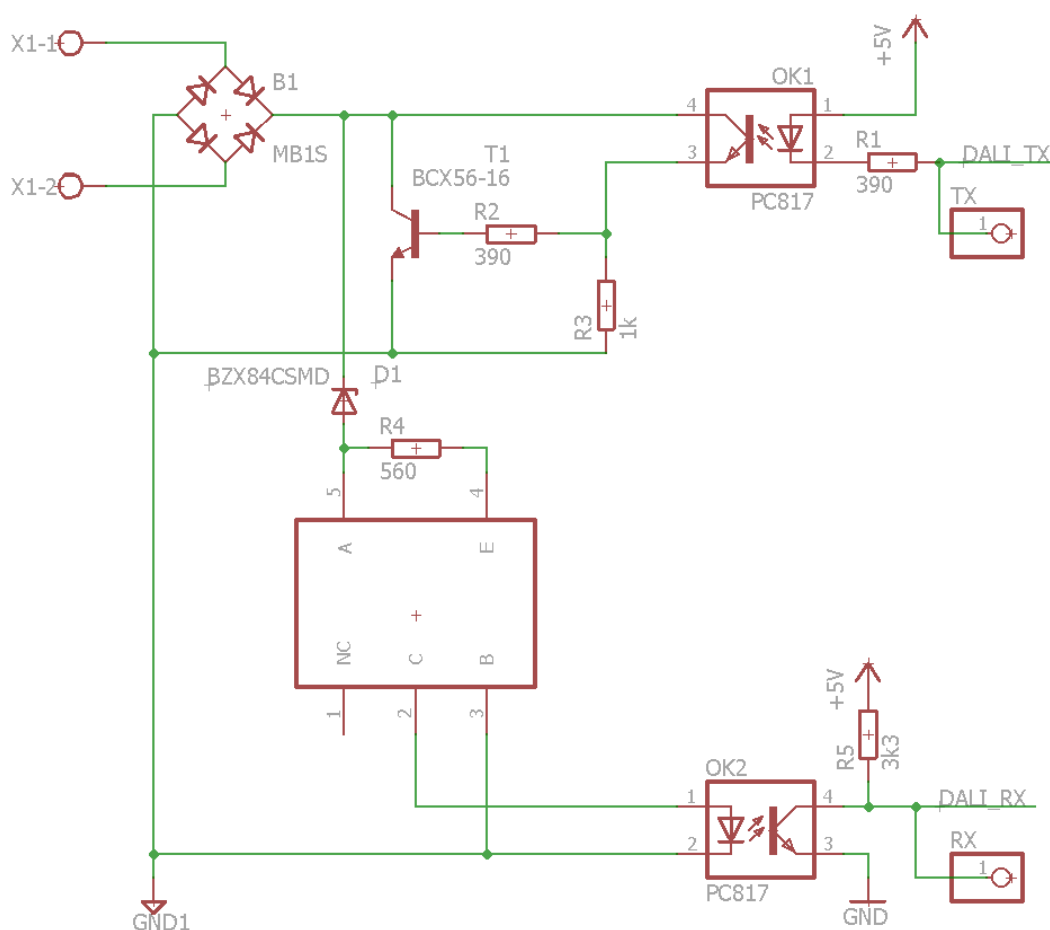
3.4 Návrh zapojení

V této části diplomové práce jsou detailně popsány jednotlivé části DALI Slave jednotek z hlediska jejich elektrického (hardwarového) zapojení. Všechna schémata byla nakreslena v aplikaci EAGLE 7.4.0 a pro účely této práce dostačovala verze nabízená zdarma.

Kompletní schémata ve formátu EAGLE a PDF jsou k nalezení v přílohách této diplomové práce.

3.4.1 Fyzická vrstva DALI

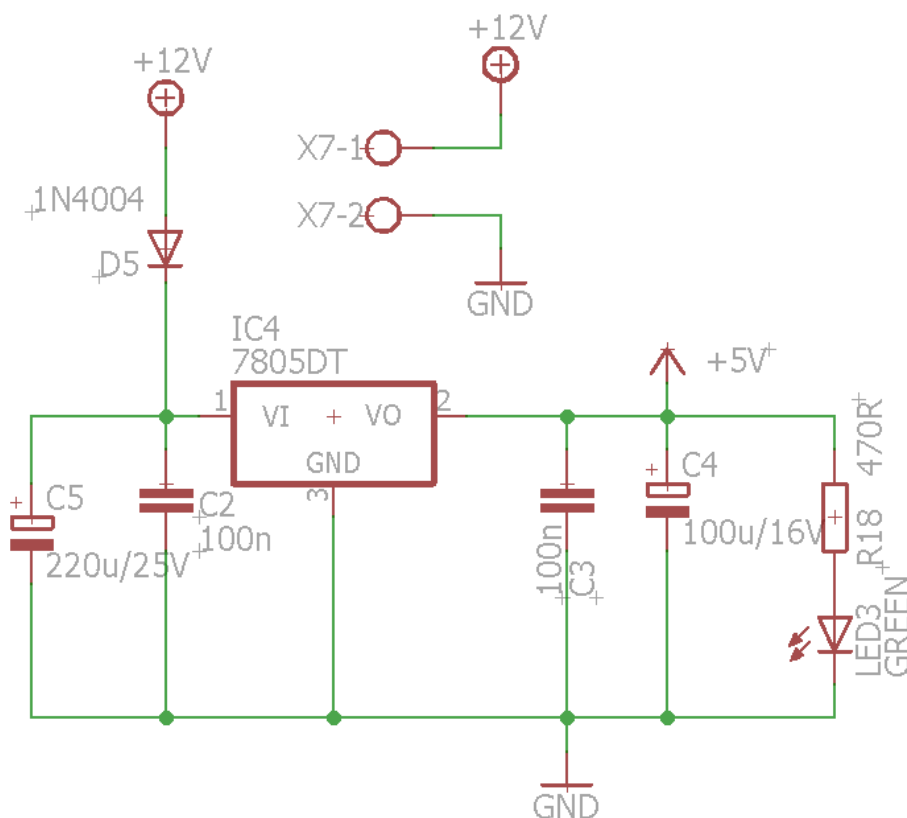
Fyzická vrstva DALI již byla popsána v sekci 2.2 Vývoj vlastních zařízení a pro účely naší DALI Slave jednotky byla převzata beze změny, pouze číslování součástek se změnilo do podoby na obrázku dole. Fyzická vrstva byla použita stejná pro vstupně-výstupní i senzorovou Slave jednotku.



Obrázek 16 – Schéma DALI fyzické vrstva na Slave jednotce

3.4.2 Napájecí část

Napájecí část je tvořena lineárním stabilizátorem napětí 7805 v SMD verzi (IC4). Vstup tohoto stabilizátoru je opatřen dvěma oddělovacími kondenzátory pro tlumení rychlých kmitů (C2) a větších proudových špiček (C5). Pro stejný účel je dvěma kondenzátory vybaven i výstup stabilizátoru IC4 (C3 a C4). Napájecí napětí pro celé zařízení je přivedeno přes svorkovnici X7 a jeho nominální hodnota je uvažována 12VDC. Celkový rozsah napájecího napětí je daný výrobcem stabilizátoru, ovšem vzhledem k tomu, že stabilizátor není nijak více chlazený, vyšší hodnoty než 15VDC nelze doporučit. Kladný pól napájení ze svorkovnice je opatřen diodou D5 v sérii, která brání poškození zařízení v případě přepólování napájecího vstupu. Napájecí část rovněž obsahuje LED diodu LED3, která indikuje, že na výstupu ze stabilizátoru je napětí. Stejně jako DALI fyzická vrstva, je i napájecí část společná pro vstupně-výstupní a senzorovou jednotku. Schéma napájecí části je uvedeno níže.



Obrázek 17 – Schéma napájecí části na DALI Slave jednotce

3.4.3 Mikrokontrolér

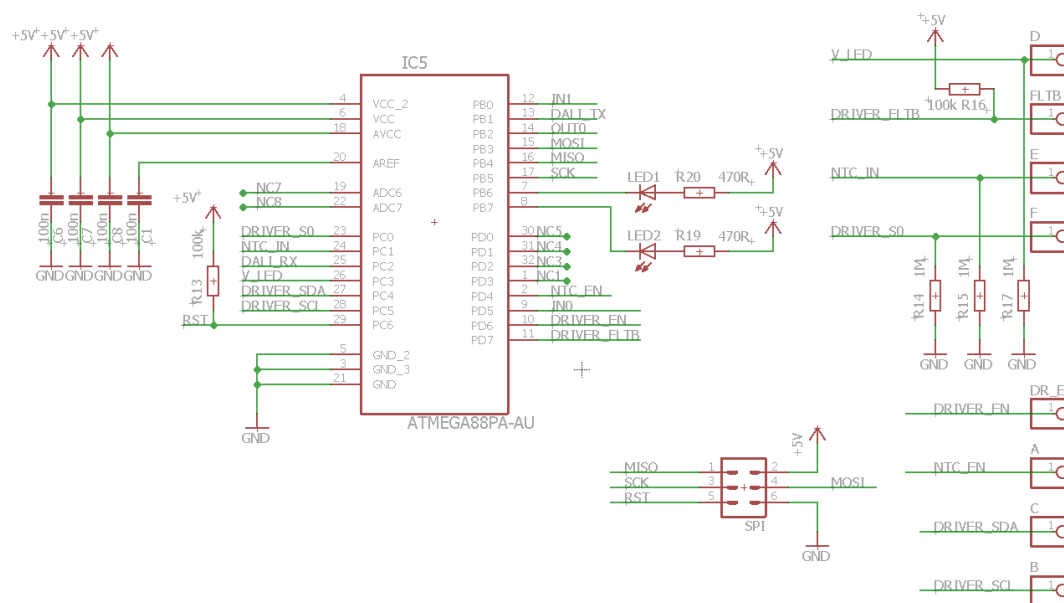
Hlavní část DALI Slave jednotky je tvořena mikrokontrolérem ATmega88 (IC5) a dalšími součástkami, které podporují jeho funkci. K napájecím pinům mikrokontroléru jsou připojeny oddělovací kondenzátory (C1, C6, C7 a C8). Indikační LED diody (LED1 a LED2) jsou rovněž připojeny přímo na mikrokontrolér. Pro účely programování firmware je v zapojení zahrnut 6-pinový konektor (SPI), který umožňuje nejen komunikaci přes sběrnici SPI, ale i speciální sériové rozhraní DebugWire.

Využití jednotlivých pinů mikrokontroléru je znázorněno v následující tabulce:

Pin	Směr dat	Signál	Popis	Poznámka
PC2	DI	DALI RX	DALI příjem dat	
PC6	DI	Reset	Reset mikrokontroléru	
PB0	DI	IN1	Digitální vstup tlačítka	pouze vstupně-výstupní jednotka
PB1	DO	DALI TX	DALI vysílání dat	
PB2	DO	OUT0	PWM výstup pro řízení zátěže	
PB3	DO	MOSI	Vstup dat pro SPI (FW update)	
PB4	DI	MISO	Výstup dat pro SPI (FW update)	
PB5	DI	SCK	Hodinový signál SPI (FW update)	
PB6	DO	LED1	výstup pro indikační LED 1	
PB7	DO	LED2	výstup pro indikační LED 2	
PD5	DI	IN0	Digitální vstup tlačítka	pouze vstupně-výstupní jednotka
PC0	AI	CURR_MEAS	Vstup měření proudu	pouze senzorová jednotka
PC1	AI	LIGHT_MEAS	Vstup měření intenzity osvětlení	pouze senzorová jednotka

Tabulka 2 – Využití pinů mikrokontroléru

Schéma části s mikrokontrolérem je přiloženo níže, pro jednoduchost pouze verze pro jednotku vstupně-výstupní.



Obrázek 18 – Schéma části s mikrokontrolérem DALI Slave jednotky

Ze schématu je patrné, že je k mikrokontroléru zavedena řada dalších signálů, které jsou však připojeny pouze k testovacím a měřícím bodům. Jedná se o signály, které byly využité v referenčním návrhu od společnosti Atmel, avšak pro účely naší DALI Slave jednotky již potřeba nebyly. Zachovány zůstaly z důvodu snadnějšího odlaďování během vývoje. Pokud by měla vzniknout nová revize zapojení Slave jednotky, mohly by již být odstraněny úplně.

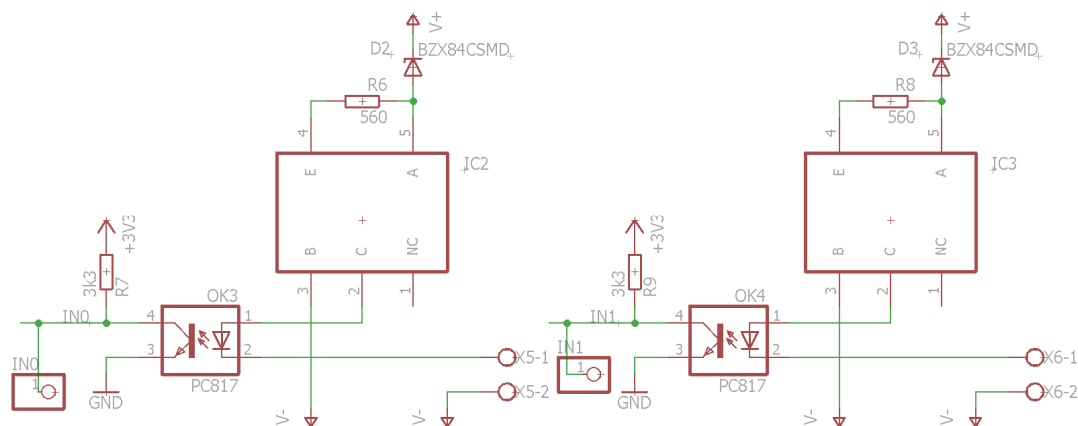
3.4.4 Vstupy

Uživatelské vstupy jsou výsadou pouze DALI Slave jednotky vstupně-výstupní a je možné na jejich vstupní svorky připojit přímo tlačítka nebo např. kontakty relé. Vstup je aktivní, pokud jsou jeho svorky vodivě propojeny a neaktivní v opačném případě.

Jelikož jsou vstupy galvanicky oddělené pomocí optočlenu, mají i samostatné napájení, které je společné s výstupem. Vstupní část optočlenů (OK3 resp. OK4) je podobně jako v případě DALI fyzické vrstvy opatřena proudovými zdroji tvořenými zenerovou diodou (D2 resp. D3) a integrovaným obvodem (IC2 resp. IC3) PSSI2021SAY. Proudové zdroje jsou zde z důvodu příliš velké variability napájecího napětí, které se může na napájecích

svorkách pro vstupy a výstupy vyskytnout, přičemž nutností je zajistit konstantní proud do LED diod uvnitř optočlenů.

Schéma vstupů je uvedeno na následujícím obrázku.



Obrázek 19 – Schéma vstupů DALI Slave jednotky vstupně-výstupní

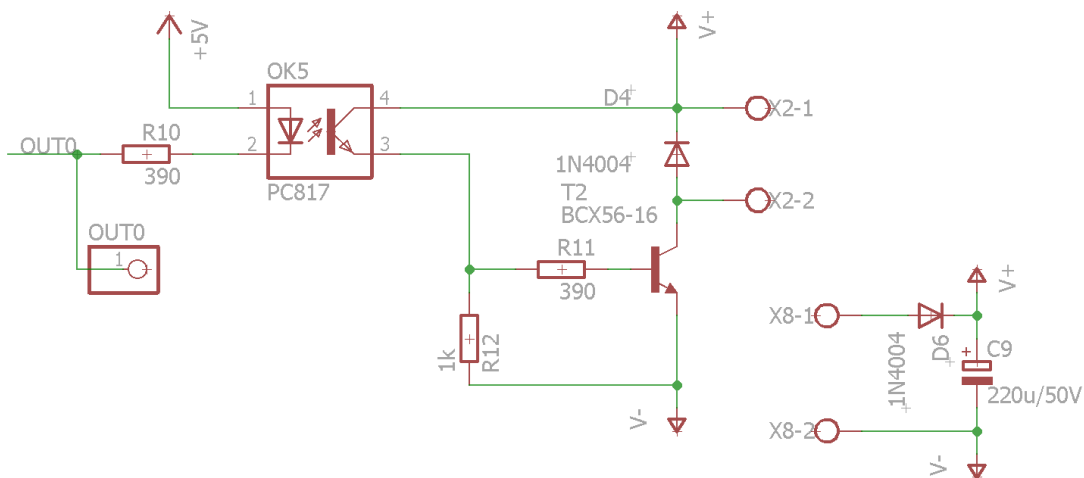
3.4.5 Výstup

Výstup je na každé DALI Slave jednotce pouze jeden. Výstup je řízený pomocí PWM z mikrokontroléru a je samostatně napájený, takže umožňuje připojení celé řady zátěží. PWM řízení umožňuje plynulou regulaci výstupního výkonu. Galvanické oddělení je zde realizováno optočlenem OK5.

Samostatné napájení je přivedeno na svorkovnici X8, která je stejně jako napájecí svorkovnice pro celý modul opatřena diodou (D6) proti přepólování. Svorkovnice X2 určena k přímému připojení zátěže. Tato svorkovnice je opatřena diodou D4 v tzv. antiparalelním zapojení kvůli eliminaci zpětných proudových špiček, pokud by byla na výstup připojena induktivní zátěž (např. cívka relé nebo elektrický motorek), i když se toto vzhledem k účelu zařízení nepředpokládá.

Proud do zátěže je řízen tranzistorem BCX56-16 (T2). Tento bipolární NPN tranzistor je schopen kolektorového proudu až 1A a ztrátového výkonu 1W.

Na schématu níže je znázorněno zapojení výstupu u vstupně-výstupní jednotky.

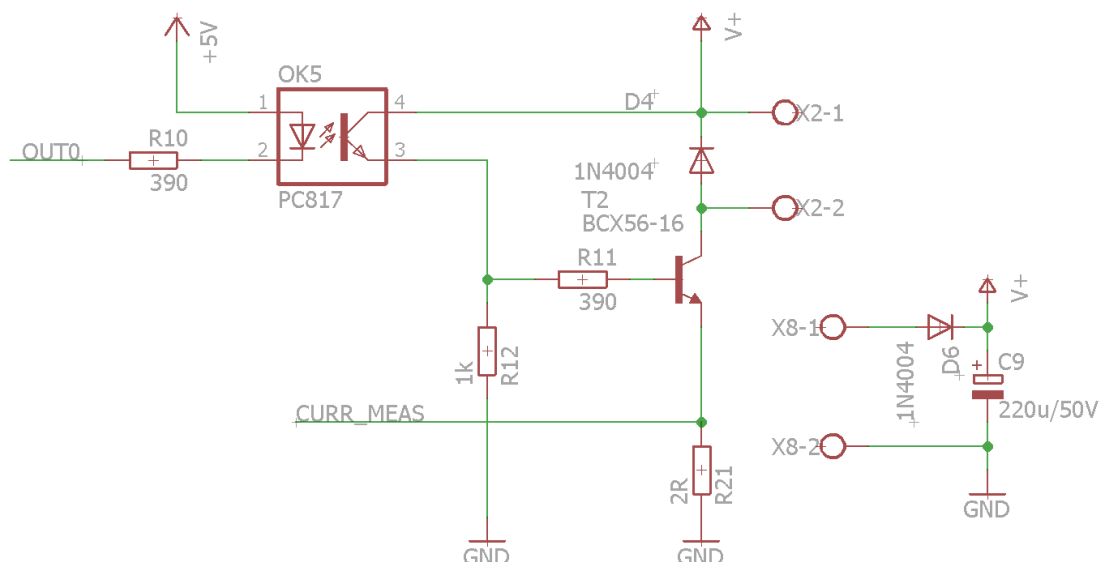


Obrázek 20 – Schéma zapojení výstupní části DALI Slave jednotky vstupně-výstupní

Slave jednotka vstupně-výstupní má výstup galvanicky oddělený. Naproti tomu jednotka senzorová je na výstupu vybavena sériovým rezistorem pro měření proudu, který protéká zátěží. Z tohoto důvodu mají obě části obvodu společnou zem a galvanické oddělení zde pozbývá smyslu. Jelikož byla senzorová Slave jednotka vyrobena až dodatečně a při jejím vývoji byla snaha co nejvíce zachovat již ozkoušenou část zapojení, byl nakonec optočlen OK5 ponechán.

Rezistor pro měření protékajícího proudu (R21) byl zvolen tak, aby při průtoku 0,5 A byl jeho napěťový úbytek 1V. Je to z toho důvodu, že interní napěťová reference pro AD převodník mikrokontroléru Atmega88 nabízí referenční napětí 1,1V a proud 0,5A je zhruba v polovině toho, co může nabídnout výstupní tranzistor T2. Jelikož zamýšlené zdroje světla jsou LED diody, nebylo očekáváno, že by výstupní proud někdy hodnotu 0,5A přesáhl. Podle Ohmova zákona tedy vychází hodnota rezistoru R21 2 ohmy.

Na obrázku níže je uvedeno schéma výstupní části pro senzorovou jednotku.



Obrázek 21 - Schéma zapojení výstupní části DALI Slave jednotky senzorové

3.4.6 Měření hodnoty okolního osvětlení

Pro účely měření intenzity okolního osvětlení je DALI Slave senzorová jednotka vybavena napěťovým děličem o dvou prvcích zapojených v sérii, kde jedním prvkem je fotorezistor a druhým obyčejný rezistor. Vstup děliče byl zapojen přímo mezi napájecí napětí mikrokontroléru a zem. Nejedná se o nejlepší řešení z hlediska přesnosti měření, ale byl kladen důraz na jednoduchost a rovněž byl brán v potaz fakt, že pro naše účely dostačuje pouze měření orientační. Pro zvýšení přesnosti měření by bylo zapotřebí použít jako vstup děliče přesnou napěťovou referenci, případně použít integrovaný obvod, který je přímo pro tento účel navržen a nabízí např. digitální výstup naměřených hodnot.

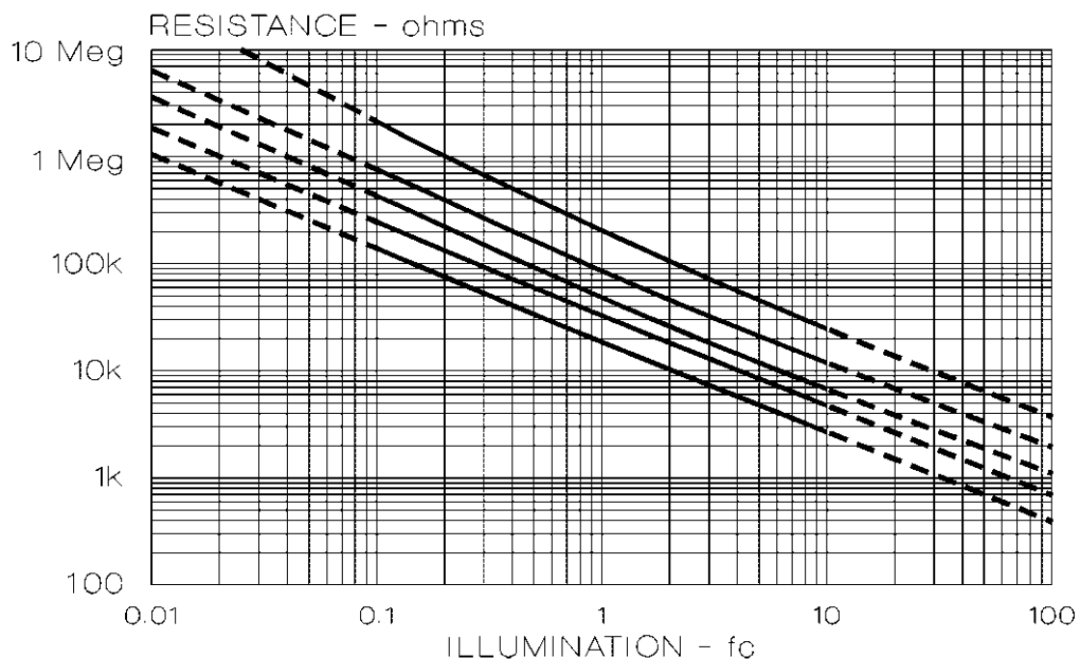
Konkrétní typ fotorezistoru byl zvolen VT93N2, jehož jmenovitá hodnota je 500kOhm při úplné tmě (19) .

Výstupem napěťového děliče je napětí mezi dvěma jeho sériově zapojenými prvky. Výstup tohoto děliče byl pak přímo zapojen na vstup AD převodníku mikrokontroléru bez žádných dalších členů, které by napěťovou úroveň jakkoliv upravovali. Při výpočtu hodnoty druhého rezistoru v děliči bylo nutné vzít v úvahu nejen napěťový rozsah vstupu AD

převodníku, ale rovněž i rozsah hodnot okolního osvětlení, ve kterých by zařízení mělo pracovat.

Fotorezistor je ze své podstaty součástka poměrně nepřesná, kde se i rozdílné kusy ze stejné série mohou výrazně lišit. Z datasheetu výrobce je patrné, že elektrický odpor fotorezistoru se může pohybovat při osvětlení 10 lux od 24kOhm do 72kOhm. Pro naše účely však bylo dostatečné počítat s hodnotou jmenovitou, která je uváděna výrobcem jako 48kOhm.

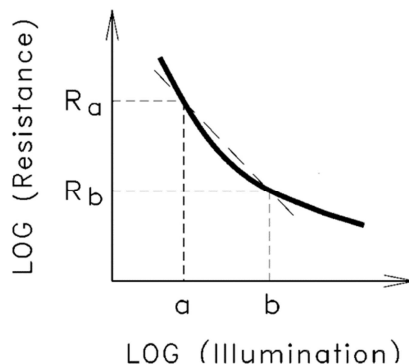
Jelikož výrobce pro daný typ součástky nenabízí přesný graf nebo výpočtový vzorec, podle kterého by se dala vyjádřit závislost elektrického odporu na intenzitě osvětlení, bylo nutné vyjít pouze z dostupných informací. Bylo tedy předpokládáno, že při zvyšující se intenzitě světla bude elektrický odpor klesat a to od hodnoty 500kOhm při úplné tmě až po hodnotu 48kOhm při intenzitě osvětlení 10lux. Rovněž bylo předpokládáno, že při dále vzrůstající intenzitě osvětlení bude elektrický odpor nadále klesat. Obecně platí pro fotorezistory logaritmická závislost intenzity osvětlení a elektrického odporu (20). Hodnota 3 v typu fotorezistoru VT93N2 určuje typ materiálu fotorezistoru. Pro tento typ materiálu je obecně stanovená závislost zobrazena v následujícím grafu:



Obrázek 22 – Graf závislosti elektrického odporu fotorezistoru na intenzitě okolního osvětlení pro materiál typu 3 (20).

Nutno podotknout, že osa hodnot osvětlení je v grafu uvedena v jednotkách foot-candle (fc), přičemž platí, že $1 \text{ fc} = 10.76391 \text{ lux}$.

Strmost křivky je nadále ovlivněna tzv. gamma citlivostí, která se dá vyjádřit jako přímka procházející dvěma specifickými body na křivce citlivosti fotorezistoru.



Obrázek 23 – Přímka gamma citlivosti (20).

Gamma citlivost je definovaná tímto vztahem:

$$\gamma = \frac{\text{Log } R_a - \text{Log } R_b}{\text{Log } a - \text{Log } b} = \frac{\text{Log}(R_a/R_b)}{\text{Log}(b/a)}$$

Obrázek 24 – Vztah vyjádření gamma citlivosti fotorezistoru (20).

kde **R_a** je hodnota elektrického odporu při intenzitě osvětlení **a**, a **R_b** je hodnota elektrického odporu při intenzitě osvětlení **b**.

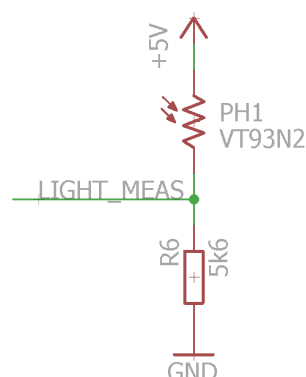
Výrobce fotorezistoru VT93N2 uvádí hodnotu $\gamma=0.9$ pro hodnoty $a=10\text{lux}$ a $b=100\text{lux}$ (19).

Bylo odhadnuto, že aby byl rozsah osvětlení 0-10lux pokryt, ale zároveň aby zůstal dostatečný prostor pro měření vyšších intenzit osvětlení, měla by být napěťová úroveň výstupu děliče při osvětlení 10lux zhruba v polovině rozsahu AD převodníku. Jelikož AD převodník pracuje s referenčním napětím 1,1V, odpovídala by tato úroveň hodnotě 0,55V. Pro zjednodušení výpočtu byla zanedbána vstupní impedance AD převodníku.

Zvažujeme-li tedy vstupní napětí děliče 5V, hodnotu horního rezistoru (fotorezistoru) 48kOhm a požadovanou hodnotu na výstupu děliče 0,55V a pokud nebereme v úvahu

vstupní impedanci AD převodníku, vychází hodnota rezistoru na 5932 Ohmů. Nejbližší hodnota v řadě je 5k6, a proto i taková byla nakonec použita.

Výsledná realizace napěťového děliče je na obrázku níže.



Obrázek 25 – Napěťový dělič sensorové DALI Slave desky pro měření intenzity okolního osvětlení

3.5 Návrh a realizace plošného spoje

Stejně jako schéma zapojení DALI Slave jednotek, byl i plošný spoj pro tu vstupně-výstupní navržen v aplikaci EAGLE 7.4.0. Jednotka sensorová se nakonec profesionálního plošného spoje nedočkala, jelikož byla navržena až dodatečně potom, co byly dva kusy jednotky vstupně-výstupní vyrobeny a také v poměrně krátké době před termínem odevzdání této diplomové práce. Zapojení jednotky sensorové bylo tedy realizováno na univerzálním plošném spoji určeném pro vývoj prototypů. Další část této kapitoly se tedy bude zabývat pouze jednotkou vstupně-výstupní.

3.5.1 Technologické požadavky

Před samotným návrhem DPS bylo nutné zhodnotit současnou situaci výrobců DPS a jejich možností. Plošný spoj DALI Slave si na výrobní možnosti příliš vysoké nároky nekladl, přesto však bylo nutné zohlednit několik požadavků, který by měl výrobce DPS splňovat:

- Oboustranná DPS s prokovenými otvory
- Nepájivá maska na obou stranách
- Servisní potisk alespoň na jedné straně
- Povrchová úprava odkrytých pájecích plošek

Výběr byl omezen pouze na tuzemské výrobce a většina z nich výše uvedené požadavky dokázala splnit. Hlavním kritériem pro výběr se tedy stala cena realizace a dodací lhůta.

Jako vhodným kandidátem se nakonec ukázala firma (20), která se specializuje na výrobu prototypových DPS v malosériových nebo i kusových množstvích. Společnost se však zabývá pouze výrobou, nikoliv však osazováním DPS. Bylo tedy ještě nutné otázku osazování vyřešit.

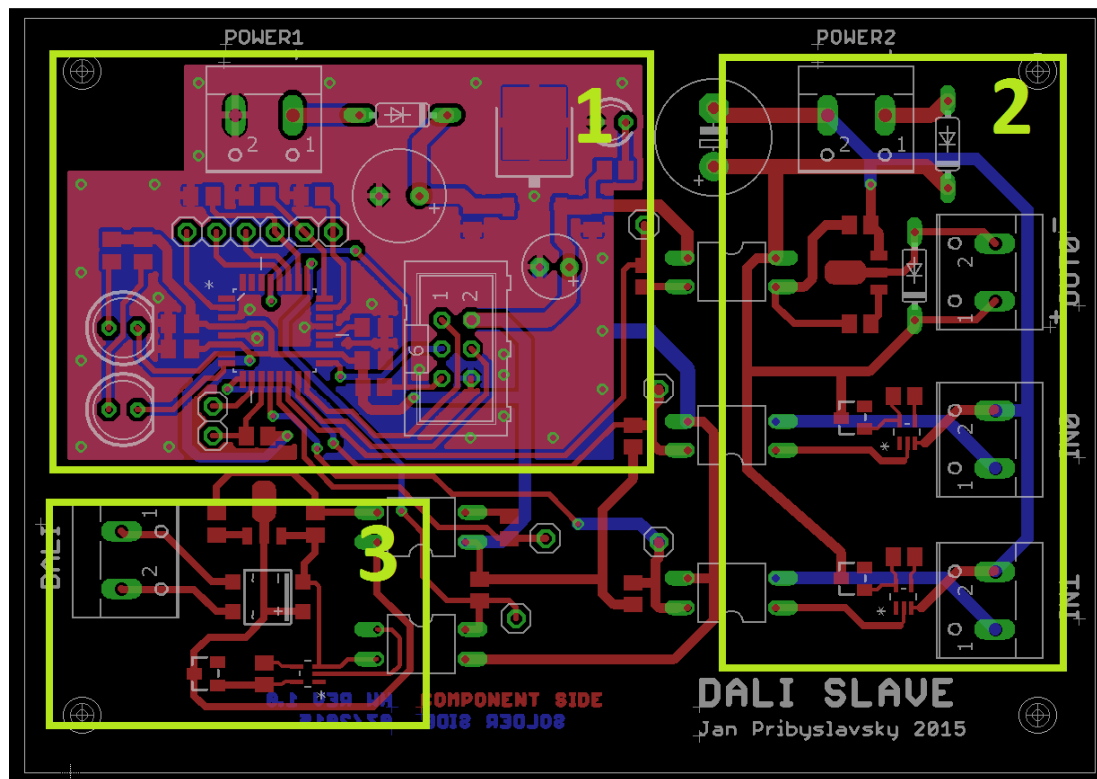
DPS byla nakonec vyrobena podle těchto parametrů:

- Oboustranná prokovená
- Tloušťka materiálu 1,5mm, tloušťka mědi 35um
- Min. tloušťka vodivé cesty 0,2mm, min. tloušťka izolační mezery 0,2mm, min průměr vrtáku 0,4mm
- Zelená nepájivá maska na obou stranách
- Bílý servisní potisk na straně součástek
- Povrchová úprava technologií HAL
- Oddělování stříháním
- Množství dva kusy

3.5.2 Návrh

Během návrhu DPS bylo nutné brát v potaz skutečnost, že celé zapojení obsahuje několik rozdílných zemí. Jendou z nich je zem mikrokontroléru, druhou zem DALI sběrnice a třetí zem vstupů a výstupů. Rozložení součástek na desce bylo poté zvoleno tak, aby byly vždy v blízkosti součástky ze stejného zemnicího bloku. Zemnicí signál mikrokontroléru byl rovněž realizován jako polygon, který na obou stranách desky zaplňuje většinu plochy.

Rozmístění jednotlivých oblastí z hlediska zemnění je znázorněno na následujícím obrázku:



Obrázek 26 – znázornění oblastí z hlediska zemnění na DPS DALI Slave board

Význam jednotlivých oblastí na obrázku je následující:

1. Zem mikrokontroléru
2. Zem vstupů a výstupu
3. Zem DALI fyzické vrstvy

Další skutečnost, se kterou bylo nutné během návrhu počítat, je řada konektorů napájecích i vstupně-výstupních, které jsou na desce umístěny. Jejich rozložení bylo zvoleno tak, aby se zařízení z mechanického hlediska snadno zapojovalo do systému, kde by mělo vykonávat svou funkci. Pro účely mechanického uchycení byla deska rovněž opatřena čtyřmi otvory o průměru 3,2 mm.

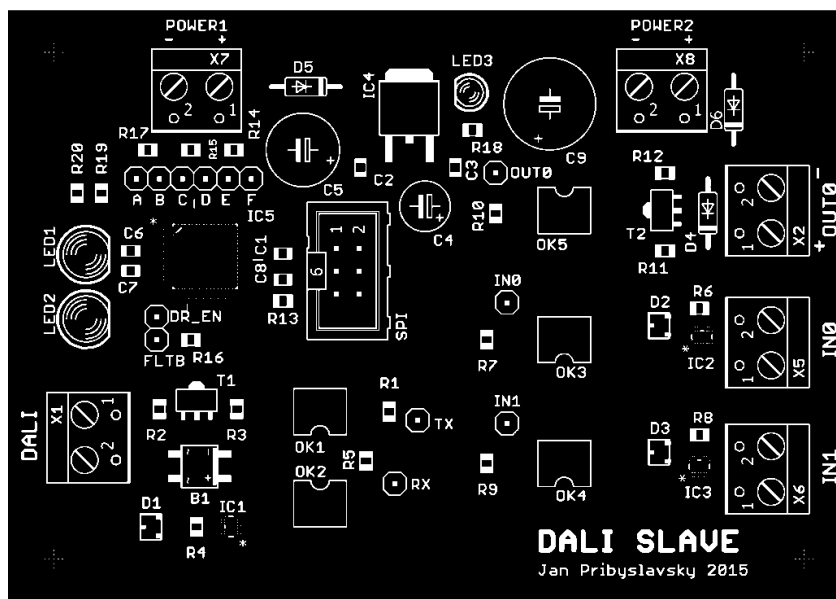
3.5.3 Osazení

Jelikož zvolený výrobce DSP nenabízí i jejich osazování, byly oba DALI Slave moduly osazeny ručně autorem této diplomové práce. Jelikož se jednalo o osazování prakticky v domácích podmínkách pouze za pomoci ruční mikropájky, bylo od samotného začátku návrhu DPS na tuto skutečnost pamatováno, a výběr pouzder součástek tomu byl přizpůsoben.

Nejjednodušším typem součástek pro ruční osazování jsou bezesporu součástky s tzv. through-hole pouzdrem. Jedná se o pouzdra s drátovými vývody, které často procházejí skrz desku. Nevýhodou through-hole součástek jsou jejich poměrně velké rozměry a rovněž i to, že ne všechny součástky (zejména integrované obvody) jsou v těchto pouzdrech dostupné.

Vzhledem k tomu, že byla pro návrh DPS použita pouze demoverze aplikace EAGLE, která limituje maximální rozměr desky na 8x10cm, bylo nutné návrh přizpůsobit tak, aby se deska do těchto rozměrů vešla. S through-hole součástkami se toto jevilo poněkud obtížné a proto byly nakonec pro návrh použity součástky pro povrchovou montáž (SMD). I přesto však byla velikost pouzder vybírána pečlivě s ohledem na ruční osazování. Jelikož byla k dispozici technologie oboustanné desky s prokovenými otvory, nebylo obtížné navrhnout desku tak, aby všechny součástky byly umístěny pouze z jedné strany desky.

Kompletní osazovací plán je uveden na obrázku níže:



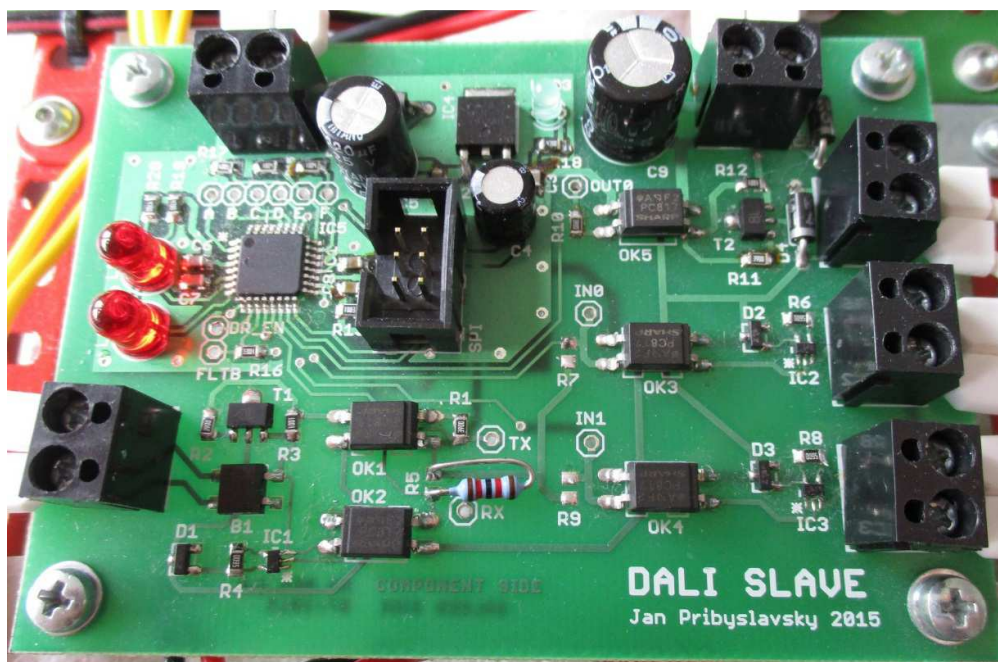
Obrázek 27 – Osazovací plán jednotky DALI Slave vstupně-výstupní

Osazení desky proběhlo podle následující rozpisky:

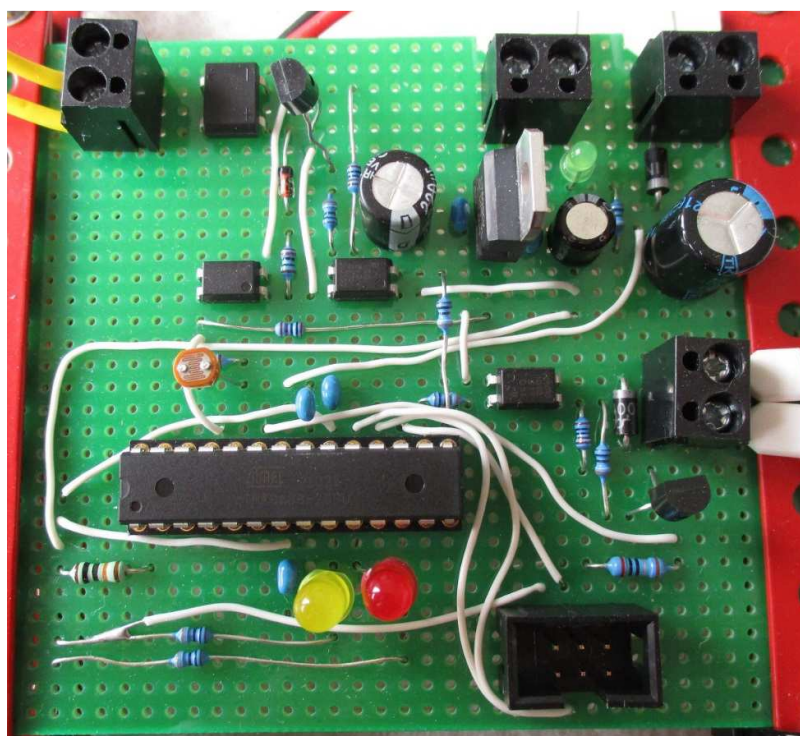
Označení	Typ	Hodnota	Pouzdro	Množství	Pozn.
C1-C3, C6-C8	Kond. Keram.	100 nF	0805	6	gme: 906-148
C4	Kond. Elyt	100 uF / 16V	RM 2,5; prům. 6	1	gme: 123-221
C5	Kond. Elyt	220 uF / 25V	RM 3,5; prům. 8	1	gme: 123-395
C9	Kond. Elyt	220 uF / 50V	RM 5; prům. 10,5	1	gme: 123-142
D1-D3	Dioda	BZX84C3V0LT1G	SOT-23-3	3	farnell: 2317469
D4-D6	Dioda	1N4007	DO41	3	farnell: 9565051
B1	Diodový můstek	MB1S	SOIC	1	farnell: 1467483
IC1-IC3	Proudový zdroj	PSSI2021SAY,115	SOT353	3	farnell: 1758034
IC4	Stabilizátor 5V	7805DT	TO252	1	farnell: 2102585
IC5	Mikrokontrolér	ATMEGA88PA-AUR	TQFP-32	1	farnell: 2425132
LED1, LED2	LED	červená	5mm kulatá	2	gme: 511-204
LED3	LED	zelená	3mm kulatá	1	gme: 511-024
OK1-OK5	Optočlen	PC817X4NSZ0F	DIP	5	farnell: 2420050
R1, R2, R10, R11	Rezistor	390 R	0805	4	gme: 901-211
R18, R19, R20	Rezistor	470 R	0805	3	gme: 901-385
R4, R6, R8	Rezistor	560 R	0805	3	gme: 901-631
R3, R12	Rezistor	1k	0805	2	gme: 901-178
R5, R7, R9	Rezistor	3k3	0805	3	gme: 901-569
R13, R16	Rezistor	100k	0805	2	gme: 901-172
R14, R15, R17	Rezistor	1M	0805	3	gme: 901-182
SPI	Konektor	MLW06G	6 pin, 2,5mm	1	gme: 800-083
T1, T2	Tranzistor	BCX5616TA	SOT-89	2	farnell: 1471209
X1, X2, X5-X8	Svorkovnice		šroubovací svorky	6	gme: 821-083

Tabulka 3 – Rozpiska součástek pro desku DALI Slave vstupně-výstupní

Na následujících obrázcích jsou k vidění hotové DALI Slave jednotky včetně jednotky senzorové, která byla zapojena pouze na univerzálním plošném spoji.



Obrázek 28 – Fotografie hotové DALI Slave jednotky vstupně-výstupní

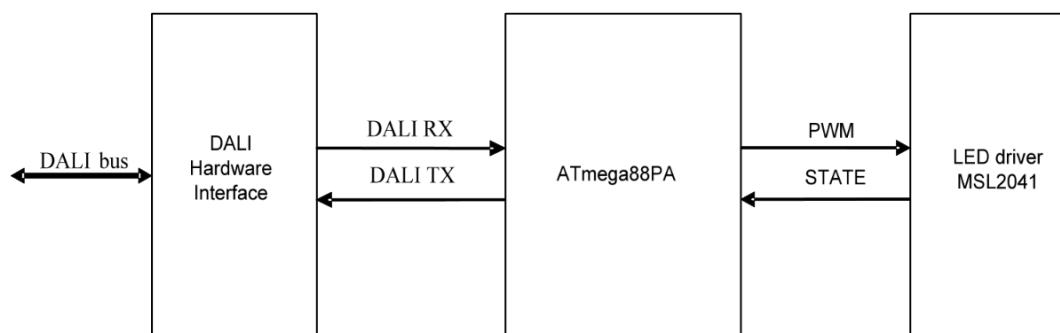


Obrázek 29 – Fotografie hotové DALI Slave jednotky senzorové

3.6 Vývoj firmware

3.6.1 Výchozí koncept

Referenční návrh pro DALI Slave jednotku od společnosti Atmel nenabízí pouze výchozí řešení z hlediska hardwarového, ale co je mnohem podstatnější, nabízí i výchozí řešení softwarové. Společnost Atmel uveřejňuje toto řešení včetně zdrojových kódů, které jsou psány poměrně přehledně a rozčleněny do několika funkčních vrstev. Referenční návrh byl koncipován podle obrázku níže – firmawre počítal s variantou, že je na výstupu zdroj světla řízen integrovaným obvodem MSL2041 od stejného výrobce. Obvod nabízí řadu pokročilých funkcí, jako např. měření proudu protékajícího zátěží a komunikaci s mikrokontrolérem přes sériové rozhraní. Využití tohoto obvodu však bylo nakonec pro naše účely zavrhnuto, zejména kvůli špatné dostupnosti této součástky a také proto, že je vyráběna pouze v TQFN pouzdru, což ji činí pro ruční osazování poměrně nevhodou. Jak už bylo zmíněno v předchozích kapitolách, na výstupu byl nakonec použit pouze spínací PNP tranzistor, jehož spínání je řízeno PWM výstupem z mikrokontroléru.



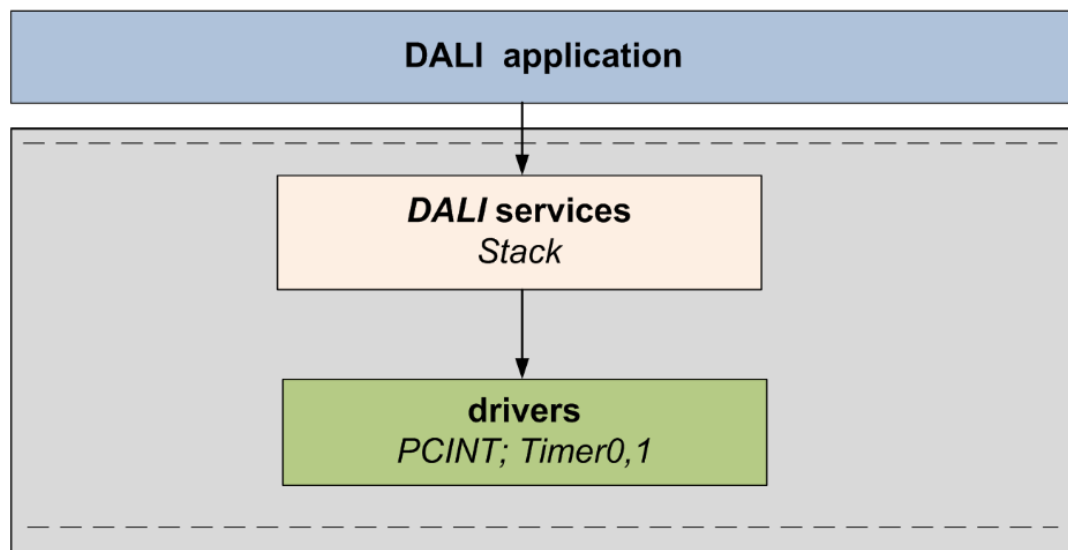
Obrázek 30 – Stručné blokové schéma referenčního návrhu DALI Slave jednotky od společnosti Atmel (18)

3.6.2 Architektura

Z hlediska architektury je celý firmware rozdělen na 3 hlavní části:

- Drivers – zpracování signálů z fyzické vrstvy
- DALI services – zpracování DALI rámců
- DALI application – realizace vlastních funkcí DALI Slave jednotky

Základní architektura, tak, jak byla převzata z referenčního návrhu, je zobrazena na následujícím obrázku:



Obrázek 31 – Architektura firmware DALI Slave jednotky (18)

3.6.3 Zpracování signálů z fyzické vrstvy (drivers)

Jelikož použitý mikrokontrolér ATMEGA88 nedisponuje žádným hardwarovým rozhraním pro přímé zpracování sériové komunikace s kódováním Manchester, je toto nutné dělat čistě softwarovou cestou. Tento problém byl vyřešen již v referenčním návrhu od společnosti Atmel a ke své realizaci používá hardwarové periférie PCINT a časovač 0.

PCINT je přerušení, které se aktivuje při každé změně hodnoty na daném vstupním pinu mikrokontroléru. V tomto případě se jedná o pin PC2, jenž je ve schématu připojen na signál DALI_RX. Signál je přímým výstupem z fyzické vrstvy DALI a pro naše účely musel být softwarově invertován z důvodu rozdílného zapojení optočlenu. Jen pro

připomenutí – fyzická vrstva zde byla převzata z desky OM13026 a nikoliv z referenčního návrhu společnosti Atmel.

Časovač 0 je nastaven tak, aby s dostatečnou frekvencí vzorkoval logickou úroveň na signálu DALI_RX resp. pinu PC2. V případě, že nastane přerušení PCINT, je hodnota pinu PC2 zaznamenána, časovač 0 spuštěn a po vyvolání přerušení od tohoto časovače je hodnota pinu opět znovu přečtena. Tímto mechanismem je postupně analyzován celý příchozí DALI paket.

Tento mechanismus je implementován v modulu **dali_bit.c** a realizován dvěmi hlavními funkcemi:

- `void dali_bit_pcint_interrupt(void)` – obsluha přerušení PCINT
- `void dali_bit_timer_interrupt(void)` – obsluha přerušení časovače 0

Při prvotním odladování tohoto softwarového modulu bylo nutné překonat některá úskalí, např. nemožnost používat breakpointy protože mechanismus je celý založený na vnějším přerušení a jakékoliv zastavení programu během přijímání paketu celý proces přesunu dat naruší. Nakonec byla použita metoda, kdy byl synchronně s některým z přerušení vygenerován krátký impuls na jednom z nevyužitých výstupních pinů, tzv. debug pinu. Signál DALI_RX byl pak společně s tímto debug pinem sledován na osciloskopu a tak bylo možné určit, kdy přesně ke kterým přerušením dochází. Např. v případě, kdy bylo nutné ověřit, zdali je časovač 0 správně nastaven a dokáže signál DALI_RX vzorkovat dostatečně rychle.

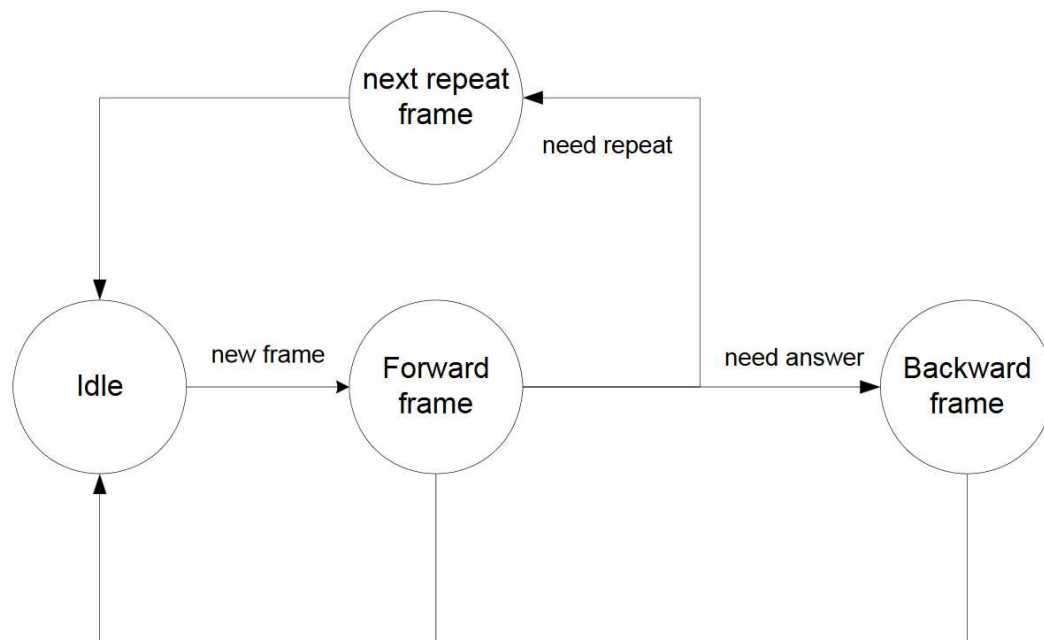
3.6.4 Zpracování rámců (DALI services)

Zpracování DALI rámců je převzato z referenčního návrhu společnosti Atmel a je implementováno jako stavový automat, jehož funkce realizující vyhodnocování stavů a přechod mezi nimi musí být periodicky volána z vyšších softwarových vrstev.

Stavový automat pro zpracování rámců se většinu času nachází v nečinném stavu (Idle) a čeká na příchozí rámeček. Pokud ten dojde, je zpracován a stavový automat v závislosti na příchozích datech přechází do dalších možných stavů – zopakování příjmu (next repeat frame), odeslání odpovědi (Backward frame) případně zpět do stavu Idle.

Norma IEC62386-100 ukládá povinnou časovou prodlevu mezi přechody jednotlivých stavů. Ta je v tomto návrhu zohledněna. Norma rovněž stanovuje, že aby byl příchozí rámeček považován za validní, musí být obdržén dvakrát v intervalu 100ms. Je to z důvodu vyloučení chybného příjmu dat. Pokud je tato podmínka splněna, je vykonán příslušný příkaz z příchozího rámce a stavový automat poté přechází do stavu Idle. Pokud se nepodaří stejný rámeček v intervalu 100ms přijmout, přechází automat do stavu Idle rovnou.

Stavový automat je vyobrazen na následujícím obrázku:



Obrázek 32 – Stavový automat zpracování DALI rámce Slave jednotkou (18)

Jak už bylo zmíněno v předchozích kapitolách, příchozí DALI rámec obsahuje dva byty. Prvním bytem je adresa a druhým data. Při zpracovávání příchozího rámce je tedy vyhodnoceno, o jaký typ adresy se jedná – adresa jednotky, adresa skupiny nebo broadcast. Oba dva byty pak dohromady tvoří kód příkazu. Pro každý takový kód je v softwaru implementovaná funkce, která se po přijetí rámce s tímto kódem vykoná.

Z implementačního hlediska je vrstva pro zpracování DALI rámců realizována v modulu **dali_frame.c** těmito hlavními funkcemi:

- `void dali_frame_process_state(void)` – funkce implementující stavový automat
- `uint8_t dali_frame_process_data(...)` – funkce realizující samotné vykonání příkazu po zpracování přijatého rámce

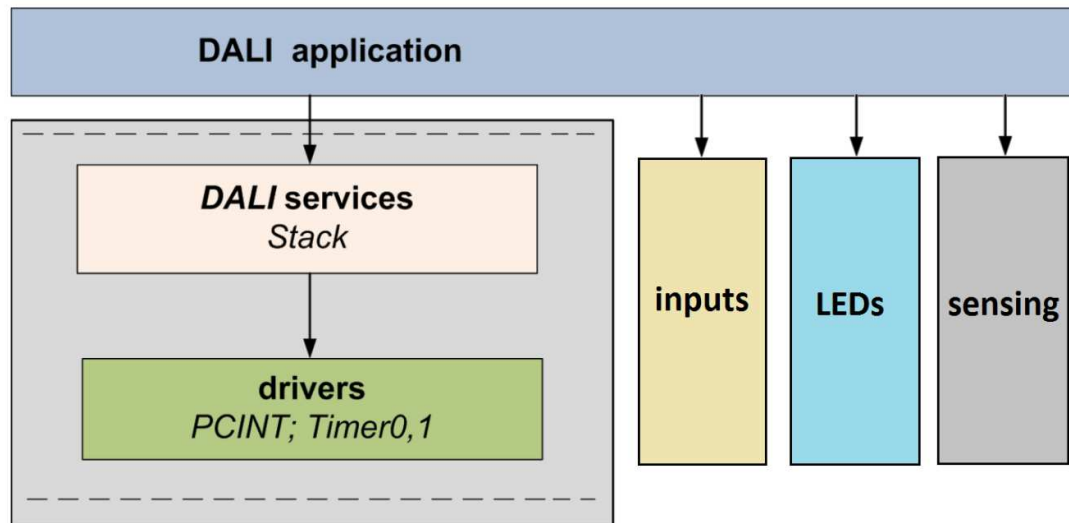
3.6.5 Aplikační vrstva (DALI application)

Pod pojmem aplikační vrstva rozumíme v naší aplikaci vše, co je z hlediska softwarové architektury na nejvyšší úrovni a co realizuje samotnou funkčnost Slave jednotky z hlediska uživatele. Pro účely realizace jednotlivých funkcí aplikační vrstvy vzniklo ještě několik nižších softwarových modulů, jejichž úkolem je zprostředkovat aplikační vrstvě přístup k hardwarovým komponentám mikrokontroléru.

Seznam těchto modulů je následující:

- LEDs.c
- inputs.c
- sensing.c

Výsledná podoba aplikace by se pak dala vyjádřit následujícím obrázkem



Obrázek 33 – Výsledná podoba architektury DALI Slave jednotky

Nutno poznamenat, že část inputs byla implementována pouze pro jednotku vstupně-výstupní, přičemž část sensing pouze pro jednotku senzorovou.

Většina funkcí aplikační vrstvy je přímo volaná příkazy, o nichž pojednává následující kapitola.

3.6.5.1 Podporované DALI příkazy

Jakýkoliv příkaz, který je zachycen během zpracování příchozích DALI rámců je předán aplikační vrstvě ke zpracování. Detailní implementace všech příkazů je k nalezení v modulu **dali_cmd.c**.

Každý příkaz má svoji hodnotu v rámci výčetového typu (enum) a jeho konkrétní implementace se nachází v rámci switch bloku, kde je na něj podle jeho enum hodnoty odskočeno.

Většina podporovaných příkazů byla převzata z referenčního návrhu společnosti Atmel a jejich seznam je uveden v tabulce níže. Pro zjednodušení vývoje našeho zařízení jsou uvedeny jen významné příkazy, jenž byly otestovány. Funkčnost kompletně všech převzatých příkazů testována nebyla z důvodu jejich značného počtu a časové náročnosti.

Kód příkazu	Název příkazu (enum)	Popis příkazu
-	samostatná funkce	Direct arc power control - přímé nastavení hodnoty pro výstup v rozsahu 0-254
0	OFF	Okamžité nastavení výstupu na hodnotu 0, přepnutí zařízení do režimu "vypnuto"
1	UP	Postupné zesvětlování po dobu 200ms pokud je zařízení v režimu "zapnuto"
2	DOWN	Postupné ztmavování po dobu 200ms pokud je zařízení v režimu "zapnuto"
3	STEP_UP	Zesvětlení o jeden krok v rozsahu 1-254 pokud je zařízení v režimu "zapnuto"
4	STEP_DOWN	Ztmavnutí o jeden krok v rozsahu 1-254 pokud je zařízení v režimu "zapnuto"
5	RECALL_MAX_LEVEL	Nastavení maximální povolené úrovně na výstup a přechod do režimu "zapnuto"
6	RECALL_MIN_LEVEL	Nastavení minimální povolené úrovně na výstup a přechod do režimu "zapnuto"
7	STEP_DOWN_AND_OFF	Ztmavnutí o jeden krok v rozsahu 0-254 pokud je zařízení v režimu "zapnuto", přechod do režimu "vypnuto", pokud je již nastavena minimální povolená úroveň výstupu
8	ON_AND_STEP_UP	Přechod do režimu "zapnuto" a zvýšení výstupní úrovně o jeden krok
16 -> 31	GO_TO_SCENE0 -> GO_TO_SCENE15	Nastaví hodnotu na výstupu podle požadované scény
32	RESET	Reset zařízení do výchozího stavu
33	STORE_ACTUAL_LEVEL_IN_THE_DTR	Uložení aktuální výstupní hodnoty do registru DTR
64 -> 79	STORE_THE_DTR_AS_SCENE0 -> STORE_THE_DTR_AS_SCENE15	Zapiše hodnotu registru DTR jako hodnotu výstupní pro danou scénu (0-15)
80 -> 95	REMOVE_FROM_SCENE0 -> REMOVE_FROM_SCENE15	Odstraní nastavení výstupní hodnoty pro danou scénu
96 -> 111	ADD_TO_GROUP0 -> ADD_TO_GROUP15	Přidá zařízení do dané skupiny
112 -> 127	REMOVE_FROM_GROUP0 -> REMOVE_FROM_GROUP15	Odstraní zařízení z dané skupiny
128	STORE_DTR_AS_SHORT_ADDRESS	Nastaví adresu zařízení podle aktuální hodnoty DTR registru
144	QUERY_STATUS	Dotaz na status zařízení
145	QUERY_CONTROL_GEAR	Dotaz na přítomnost zařízení. Zařízení vždy odpovídá 0xFF.
146	QUERY_LAMP_FAILURE	Dotaz zda-li zařízení hlásí chybu.
147	QUERY_LAMP_POWER_ON	Dotaz zda-li je zařízení v režimu "zapnuto"
148	QUERY_LIMIT_ERROR	Dotaz na limit error
149	QUERY_RESET_STATE	Dotaz zda-li je zařízení v reset stavu

150	QUERY_MISSING_SHORT_ADDRESS	Dotaz zda-li má zařízení chybějící nastavení adresy
151	QUERY_VERSION_NUMBER	Dotaz na verzi zařízení
152	QUERY_CONTENT_DTR	Dotaz na aktuální hodnotu registru DTR
153	QUERY_DEVICE_TYPE	Dotaz na typ zařízení
154	QUERY_PHYSICAL_MINIMUM_LEVEL	Dotaz na minimální fyzickou úroveň na výstupu
155	QUERY_POWER_FAILURE	Dotaz na power failure
156	QUERY_CONTENT_DTR1	Dotaz na obsah registru DTR1 - Pro vstupně-výstupní jednotku vždy vrací status tlačítka 0. Pro senzorovou vždy vrací hodnotu proudu zátěží.
157	QUERY_CONTENT_DTR2	Dotaz na obsah registru DTR2 - Pro vstupně-výstupní jednotku vždy vrací status tlačítka 1. Pro senzorovou vždy vrací hodnotu intenzity okolního osvětlení.
160	QUERY_ACTUAL_LEVEL	Dotaz na aktuální hodnotu na výstupu
161	QUERY_MAX_LEVEL	Dotaz na maximální povolenou úroveň na výstupu
162	QUERY_MIN_LEVEL	Dotaz na minimální povolenou úroveň na výstupu
163	QUERY_POWER_ON_LEVEL	Dotaz na výstupní úroveň, která je nastavena po zapnutí zařízení
164	QUERY_SYSTEM_FAILURE_LEVEL	Dotaz na výstupní úroveň, která je nastavena během chyby
176 -> 191	QUERY_SCENE_LEVEL0 -> QUERY_SCENE_LEVEL15	Dotaz na výstupní úroveň pro danou scénu
229	SET_INDICATION_LED_0	Rozsvícení indikační LED 0
230	CLEAR_INDICATION_LED_0	Zhasnutí indikační LED 0
231	SET_INDICATION_LED_1	Rozsvícení indikační LED 1
232	CLEAR_INDICATION_LED_1	Zhasnutí indikační LED 1

Tabulka 4 – Seznam podporovaných příkazů DALI Slave jednotky

3.6.5.2 Řízení výstupu

Z hlediska softwaru je výstup řízen přímým nastavením PWM časovače realizující PWM výstup na příslušném pinu mikrokontroléru. Funkčnost je implementována v modulu **dali_cmd.c** a funkci `void dali_cmd_set_pwm_output(uint8_t dali_value)`.

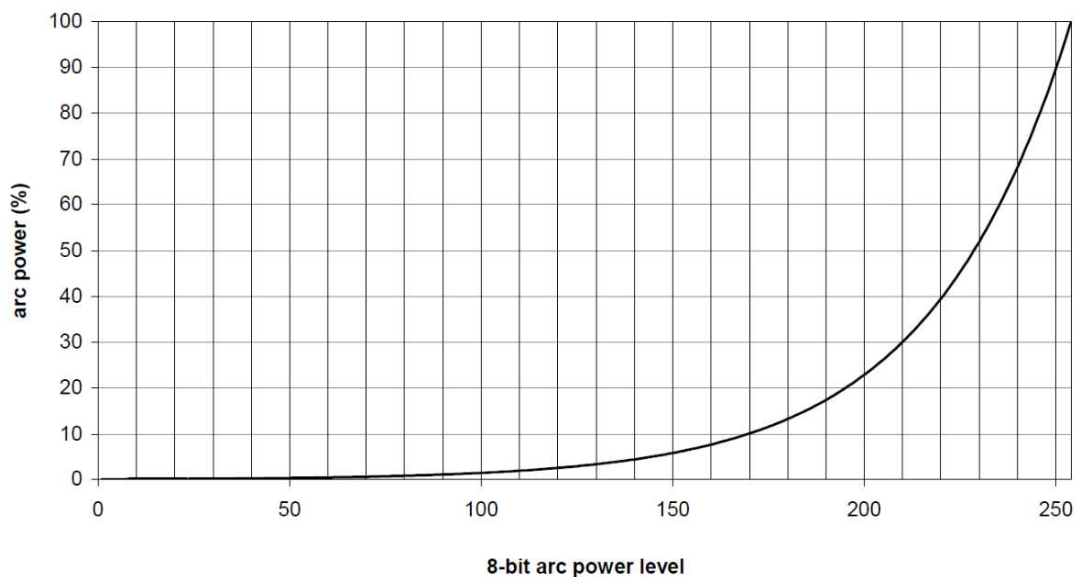
Funkce přijímá jediný parameter, 8 bitovou hodnotu požadované výstupní úrovně. Je třeba mít napaměti, že závislost je zde logaritmická. Jelikož je zamýšlenou zátěží na výstupu světelný zdroj, konkrétně LED dioda, je závislost přizpůsobena citlivosti lidského oka na jas.

Hodnoty pro výstup nejsou přímo počítány, ale jsou načteny z tzv. LUT tabulky, která je umístěná v paměti EEPROM. Hodnoty byly předpočítány předem podle vztahu, jenž je definován přímo DALI standardem.

$$X(n) = 10^{\frac{n-1}{253/3} - 1} \quad \left| \frac{X(n) - X(n+1)}{X(n)} \right| = \text{constant} = 2.8 \%$$

Kde X je úroveň výstupního výkonu v % a n je požadovaný krok v rozsahu 1-254, kde krok 1 odpovídá výstupnímu výkonu 0,1% a krok 254 100%.

Závislost výstupního výkonu na hodnotě n je znázorněna na následujícím grafu:



Obrázek 34 – Graf závislosti výstupního výkonu na 8bit hodnotě požadovaného kroku (18)

3.6.5.3 Řízení indikačních LED diod

Řízení indikačních LED diod je implementováno v modulu **LEDS.c**. Před samotným používáním celého modulu je nutné alespoň jednou zavolat jeho funkci `void LEDs_init(void)`, která celý modul inicializuje a nastaví patřičné výstupní piny mikrokontroléru pro jejich správnou funkci. Modul si udržuje interní proměnnou `LEDs_state`, která plní funkci registru o tom, která indikační LED by měla svítit a která nikoliv. Každý bit této proměnné odpovídá právě jedné LED a je tedy teoreticky možné pracovat až s 8 indikačními LED, byť hardwarově jsou přítomné pouze dvě.

Skutečná konfigurace fyzických LED je definována výčtovým typem jako:

```
typedef enum {  
    LED0,  
    LED1  
} LEDs_t;
```

Jedná se zároveň o datový typ, který je používán vždy, kdy dochází k volání některých funkcí pracujících s indikačními LED diodami.

Modul dále implementuje tyto funkce:

- `void process_LEDs(void)` – funkce starající se o řízení hodnot na výstupech pro indikační LED. Jako jediná přímo přistupuje na výstupní piny mikrokontroléru. Její jeden průběh způsobí, že jsou indikační LED fyzicky nastaveny podle odpovídajících úrovní příslušných bitů v proměnné `LEDs_state`. Pro správnou funkčnost v reálném čase musí být tato funkce volána periodicky.
- `void set_LED(LEDs_t led)` – funkce nastaví požadovanou LED jako aktivní (rozsvícenou)
- `void clear_LED(LEDs_t led)` – funkce nastaví požadovanou LED jako neaktivní (zhasnutou)
- `void set_LEDs_status(uint8_t status)` – funkce nastaví náraz celou proměnnou `LEDs_state`, umožňuje tedy rozsvítit nebo zhasnout několik indikačních LED v jednom kroku.

3.6.5.4 Zpracování signálů ze vstupů

Zpracování signálů z digitálních vstupů je implementováno v modulu **inputs.c** a jeho funkčnost byla navržena s ohledem na to, že jsou na digitálních vstupech tlačítka. Modul funguje podobně, jako modul LEDs.c, který zpracovává indikační LED diody. Opět zde nalezneme inicializační funkci `void inputs_init(void)`, jenž musí být před používáním zbytku modulu alespoň jednou zavolána. Funkce se stará o správné nastavení HW periférií, zejména vstupních pinů. Modul si rovněž udržuje svoji vnitřní proměnnou `inputs_states`, kde každý její bit odpovídá jednomu z digitálních vstupů. Zde jsou opět fyzicky přítomny pouze dva.

Datový typ pro práci s jednotlivými tlačítky je zde definován jako:

```
typedef enum {  
    BUTTON_0,  
    BUTTON_1  
}Buttons_t;
```

Další funkce modulu jsou potom tyto:

- `void process_inputs(void)` – funkce realizující aktualizaci proměnné `inputs_states`. Pokud je během vykonávání této funkce kterékoliv tlačítko uživatelem zmáčknuto, je příslušný bit v této proměnné nastaven. Funkce musí být volaná periodicky s dostatečně krátkou periodou pro zajištění funkčnosti v reálném čase.
- `bool inputs_pending(void)` – funkce vrátí true nebo false v závislosti, zda-li je některý z bitů v proměnné `inputs_states` nastaven. Zpravidla je možné tuto funkci použít k dotazu, zda-li došlo k události stisku některého tlačítka.
- `uint8_t read_inputs(void)` – funkce vrátí hodnotu proměnné `inputs_states`.
- `void clear_inputs(void)` – funkce nastaví všechny vstupy tak, jakoby nedošlo k jejich aktivaci. Tedy hodnotu proměnné `inputs_states` rovnu nule.
- `bool read_input(Buttons_t button)` – funkce vrátí true nebo false v závislosti na tom, zda-li byl příslušný vstup aktivován (tlačítko stisknuto).
- `void clear_input(Buttons_t button)` – funkce vynuluje příslušný bit v proměnné `inputs_states` odpovídající danému tlačítku.

Jelikož DALI Slave jednotka nemůže sama osobě iniciovat komunikaci na DALI sběrnici, a tudíž ani nemůže sama vyslat informaci o tom, že právě došlo ke stisku tlačítka, musí se

Master neustále a opakovaně na stav tlačítek dotazovat. Z tohoto důvodu se události stisku tlačítka zaznamenávají do proměnné `inputs_states`, aby nedošlo ke strátě události v případě, že by se Master dotazoval v příliš dlouhých intervalech a uživatel stiskl tlačítko krátce mezi těmito intervaly. Tato metoda se běžně nazývá *polling*.

Aby bylo možné zaznamenat událost stisku tlačítka opakovaně, je příslušný bit v proměnné `inputs_states` po jeho přečtení vynulován. Toto realizují funkce pro vykonání příkazů `QUERY_CONTENT_DTR1` a `QUERY_CONTENT_DTR2` v modulu **`dal_i_cmd.c`**.

3.6.5.5 Měření intenzity okolního osvětlení

Měření intenzity okolního osvětlení je pouze funkcí DALI Slave jednotky senzorové a zpravidla využívá interní AD převodník mikrokontroléru. Funkčnost je implementována v souboru `sensing.c` těmito funkcemi:

- `void ADC_init(void)` – Funkce zajišťující inicializace modulu. Před používáním ostatních funkcí modulu musí být zavolána alespoň jednou. Funkce zajistí povolení běhu AD převodníku, dále nastaví multiplexor vstupů na pin PC1, který je fyzicky propojený s výstupem odporového děliče s fotorezistorem. Funkce rovněž nastaví zdroj referenčního napětí pro AD převodník. V tomto případě je to interní reference 1,1V.
- `void process_ADC(void)` – Funkce provádějící samotné měření. Každé zavolání funkce provede právě jeden převod a výsledek uloží do statické proměnné modulu - `adc_value`.
- `uint8_t GetLevel(void)` – Funkce určená k přečtení požadované hodnoty založené na měření intenzity okolního osvětlení. Funkce načte hodnotu z proměnné `adc_value` a převede ji podle jednoduchého algoritmu na hodnotu v rozsahu 0-255. Tato hodnota může být dále přímo předána jiné DALI jednotce za účelem úpravy výkonu na výstupu.

3.6.5.6 Měření výstupního proudu do zátěže

Tato funkčnost DALI Slave jednotky senzorové byla zpracována pouze ve formě konceptu a dostalo se jí pouze hardwarového zapojení. Softwarová implementace již nebyla realizována.

3.6.6 Hlavní smyčka programu

Hlavní smyčka programu je implementována jako nekonečný cyklus while přímo ve funkci `int main(void)` v modulu `dali_main.c`. Jednotlivé funkce zajišťují běh zařízení v reálném čase.

Smyčka je implementována tímto způsobem:

```
while (1) {  
    dali_process_power_on();  
    dali_frame_process_state();  
    dali_process_interface_failure();  
    process_inputs();  
    process_LEDs();  
}
```

4 PRAKTICKÁ REALIZACE TESTOVACÍ FIXTURY

4.1 Účel testovací fixtury

Jednotka DALI Slave nenabízí příliš mnoho možností, pokud bychom ji chtěli testovat nebo provozovat samostatně. Až na několik základních testů, jako např. změření napětíových úrovní na napájecích signálech, případně na výstupech, se o její správné funkčnosti nedozvíme téměř nic.

Pro účely komplexního testování DALI Slave jednotek, stejně tak i pro demonstraci funkčnosti systému řízeného s využitím DALI sběrnice, vznikla tzv. testovací fixtura.

Pojem „testovací fixtura“ vychází z angl. „test fixture“ a je hojně využíván v oblasti softwarového a systémového inženýrství. Pojem nemá v češtině ekvivalent, avšak v této práci bude pro svoji výstižnost využíván.

4.2 Popis testovací fixtury

Testovací fixtura je elektronický systém, který je vybaven několika DALI Slave jednotkami, jednou DALI-Master jednotkou a řadou dalších prvků, jako např. tlačítka a světelné zdroje.

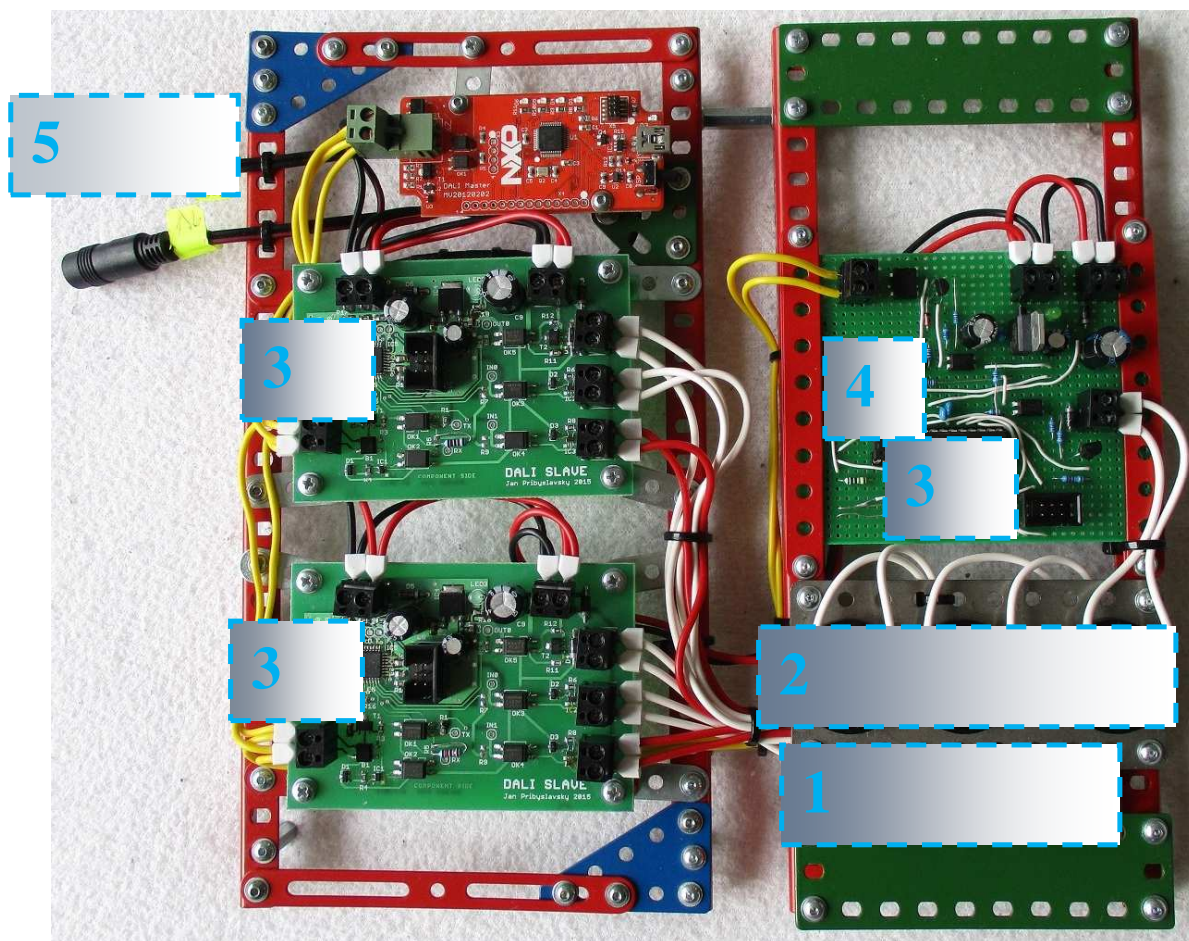
Celý systém je zapojený tak, aby jednotlivé DALI Slave jednotky mohly komunikovat se svým okolím, jakoby se jednalo o reálnou situaci využití sběrnice DALI. Mechanická část fixtury je tvořena kovovým rámem ze stavebnice Merkur.

Testovací fixtura se skládá z těchto komponentů:

- 2x vstupně-výstupní DALI Slave jednotka
- 1x senzorová DALI Slave jednotka
- 1x DALI-Master jednotka
- 24VDC adaptér pro napájení DALI sběrnice
- 9VDC adaptér pro napájení DALI jednotek
- 3x světelný zdroj
- 4x tlačítko

Testovací fixtura umožňuje zejména následující:

- Uživatelský vstup pomocí 4 samostatných tlačítek (obr. - 1)
- Výkonové řízení 3 světelných zdrojů (9V LED „žárovky“) (obr. - 2)
- Indikace uživatelsky nastavitelných událostí pomocí 6 indikačních LED diod (obr. - 3)
- Detekce jedné poruchy (odpojení zátěže) na jednom světelném zdroji
- Snímání intenzity okolního osvětlení (obr. – 4)
- Napájení DALI sběrnice (obr. – 5)



Obrázek 35 – Fotografie hotové testovací fixtury

4.3 Volba jednotky DALI-Master

Jednotka DALI-Master je modul, bez kterého se systém využívající sběrnici DALI neobejde. Jedná se o klíčový člen v komunikaci po sběrnici, který jako jediný iniciuje veškeré přesuny dat. Na trhu lze nalézt celou řadu zařízení, která funkci jednotky DALI-Master mohou zastupovat, a která disponují širokou škálou dalších funkcí. Sofistikované (a mnohdy i poměrně nákladné) moduly nabízí značný stupeň autonomie a rozhraní pro jiné sběrnice. S pomocí takových modulu je možné budovat rozsáhlé sítě inteligentních budov, které se zdaleka neomezují jen na maximální počet 64 slave jednotek na jednom okruhu DALI sběrnice.

Pro účely naší testovací fixtury však nebylo nutné hledat takto pokročilý modul, jelikož je zde velmi omezený počet slave jednotek, fixtura není součástí žádného většího systému a stejně tak i autonomie master jednotky je jen velmi malá, protože celý systém spolupracuje s PC-aplikací, která většinu řídicích funkcí může zastoupit.

Jako hlavní kritéria pro výběr DALI-Master jednotky byly vybrány tyto body:

- Cenová dostupnost (horní limit 2000,- Kč)
- Standardní rozhraní pro komunikaci s PC (např. USB, RS232, Ethernet atd...)
- Možnost otevřeného vývoje na straně PC (zařízení není součástí nějakého uzavřeného komerčního systému)
- Dostupnost ovladačů pro Windows 7
- Dostupnost knihoven pro platformu .NET (platforma řídící PC-aplikace)

Jelikož bylo od začátku o Master jednotce uvažováno jako o hotovém modulu, na němž nebude stráven žádný čas vývojem dodatečné funkcionality, nebylo zde přistupováno k žádným pokročilejším metodám výběru, jako tomu bylo např. v případě výběru konceptu DALI Slave jednotky. Nakonec bylo rozhodnuto vybrat první modul, který vyhoví požadavkům v předchozím odstavci.

Jako vhodný kandidát pro master jednotku se nakonec jevil modul OM13046 od společnosti NXP. Jedná se prakticky o převodník mezi sběrnici DALI a rozhraním USB, který je založený na mikrokontroleru LPC134x. Zařízení je napájeno přímo z USB a pro komunikaci s PC využívá třídu HID (Human Interface Device), což značně zjednodušuje

vývoj software na straně PC, jelikož je pro toto zařízení možné využít nativní ovladače operačního systému Windows a platformy .NET.

Po hardwarové stránce zahrnuje modul OM13046 fyzickou vrstvu DALI sběrnice s galvanickým oddělením pomocí optočlenů. Modul sám o sobě nedokáže DALI sběrnici napájet a toto je nutné zajistit samostatným proudovým zdrojem. Na plošném spoji jednotky master je osazeno několik LED diod, které indikují různé stavy samotného zařízení a sběrnice. Deska rovněž nabízí expanzní konektor pro další uživatelské funkce.

Výrobce modulu, společnost NXP, uveřejnil pro toto zařízení veškeré podklady, včetně zdrojových kódů. Tyto však pro účely této práce nebyly nijak modifikovány.



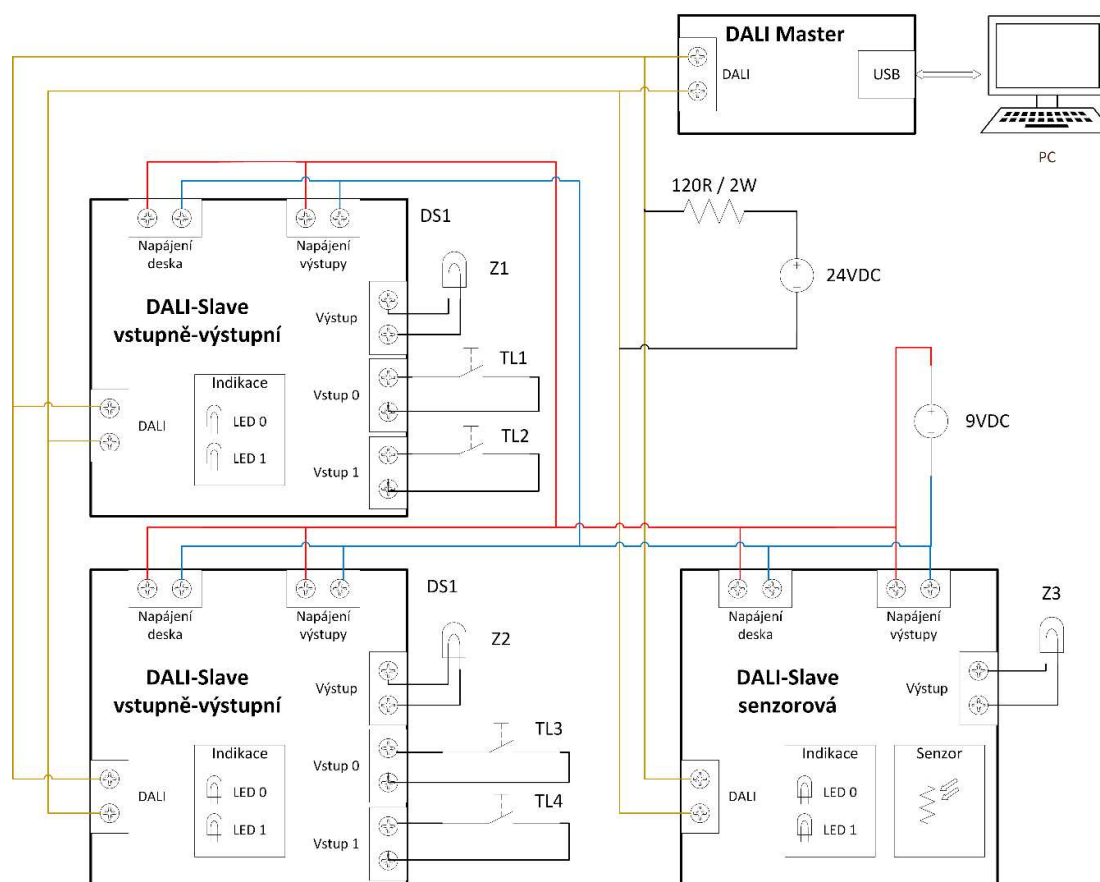
Obrázek 36 – DALI Master modul od společnosti NXP

4.4 Zapojení testovací fixtury

V této sekci je uvedeno detailní zapojení testovací fixtury. Při zapojování byly využity vodiče – lanka o průřezu 0,5 mm² v barvách červená, černá, bílá a žlutá.

Červeně a černě byly zapojeny napájecí vodiče (červená +, černá -). Červená byla rovněž použita pro zapojení tlačítek TL2 a TL4, která mají odpovídající barvu. DALI sběrnice byla zapojena žlutými vodiči, přičemž vše ostatní bílými.

Pro snadnou dostupnost byly všechny součástky a díly testovací fixtury nakoupeny v pobočce tuzemského prodejce GM Electronic.



Obrázek 37 – Schéma zapojení tetovací fixtury

Pro úplnost je níže uvedena rozpiska všech konstrukčních dílů testovací fixtury.

Součástka	Ks	Hodnota	Obj. č. GME
LED žárovka bílá	3	LED žárovka E10 teplá bílá	518-086
objímka	3	E10 1153-500	070-102
tlačítko bílé	4	P-PB303B WHITE	630-100
tlačítko červené	4	P-PB303B WHITE	630-100
dutinka na kabel	30	DK8x2x0,5-W	834-079
vidlice na kabel	10	KSS YF1.25-3	834-170
stahovací pásek	30	KSS CV-100 x 2,5mm	656-511
distanční sloupek M3x20	10	DI5M3X20	623-153
distanční sloupek M3x10	10	DI5M3X10	623-151

šroub M3x6	20	SKM3X6	662-005
šroub M3x16	20	SKM3X16	662-007
podložka 3mm	40	SKM3UK	662-002
matice M3	20	SKM3K	662-001
kabel 0,5mm černý	2m	CYA 1x0,5 CERNY H05V-K	651-276
kabel 0,5mm červený	2m	CYA 1x0,5 CERVENY H05V-K	651-285
kabel 0,5mm bílý	2m	CYA 1x0,5 BÍLÝ H05V-K	651-010
kabel 0,5mm žlutý	2m	CYA 1x0,5 ZLUTY H05V-K	651-275

Tabulka 5 – rozpiska součástí testovací fixtury

4.5 Konfigurace jednotlivých modulů

Aby se mohla testovací fixtura chovat jako reálný systém řízený DALI sběrnici, bylo třeba jednotlivé Slave jednotky nakonfigurovat. Konfigurace byla provedena přímo ve zdrojovém kódu jednotlivých Slave jednotek a po přeložení a nahrání tohoto kódu již nelze změnit.

Konfigurace jednotek spočívá především v definování adres a skupin pro jednotlivé moduly a byla provedena tímto způsobem:

Jednotka	Typ	Adresa	Skupina
DS1	vstupně-výstupní	1	0
DS2	vstupně-výstupní	2	0
DS3	senzorová	3	1

Tabulka 6 – konfigurace modulů na testovací fixtuře

5 ŘÍDICÍ A KONFIGURAČNÍ PC-APLIKACE

5.1 Popis PC-aplikace

PC aplikace vznikla za účelem řízení a konfigurace celého systému využívajícího DALI sběrnici. Smyslem aplikace je poskytnout uživateli univerzální možnosti, jak celý systém spravovat.

Hlavní funkčnost PC aplikace je možné shrnout do těchto bodů:

- Umožnit uživateli konfigurace a řízení systému DALI sběrnice prostřednictvím grafického rozhraní
- Nabídnout uživateli možnosti pro správu konfiguračních dat jednotlivých DALI zařízení na sběrnici
- Realizovat komunikaci s DALI Master jednotkou
- Realizovat řídicí funkce systému

5.2 Výchozí koncept

Výchozí koncept se z velké části odvíjí od použité DALI Master jednotky, kterou je modul OM13046 od společnosti NXP. Výrobce tohoto modulu k němu nabízí i řídicí PC aplikaci včetně zdrojových kódů. Výhodou je zde skutečnost, že tato aplikace již plně implementuje komunikaci s DALI Master jednotkou prostřednictvím sběrnice USB a protokolu HID. Rovněž je zde poměrně rozsáhlým způsobem implementována obsluha sběrnice DALI.

Hlavním úkolem při vývoji PC aplikace pro tuto diplomovou práci bylo tedy rozšířit výchozí aplikaci od společnosti NXP o řídicí funkce a možnost správy konfiguračních dat.

Výchozí aplikace byla převzata z DALI Development Kit Revision 2 (21).

5.3 Realizace PC-aplikace

5.3.1 Použité technologie

PC aplikace je téměř výhradně napsána v jazyce C# nad platformou .NET. Jako vývojové prostředí bylo zvoleno Microsoft Visual Studio Express 2013. Původní myšlenkou bylo využití návrhového vzoru MVC, avšak toto se nakonec ukázalo jako poněkud problematické, jelikož samotný výchozí koncept se tohoto návrhového vzoru nedržel a jeho přizpůsobení by znamenalo vynaložení poměrně značného úsilí. Z hlediska návrhového vzoru se tedy aplikace nedrží žádných striktních pravidel.

Pro uchovávání informací o jednotlivých DALI Slave jednotkách, ale i o nastavení řídicích funkcí využívá aplikace ve velké míře jazyk XML, včetně jeho rozšíření XML Schema.

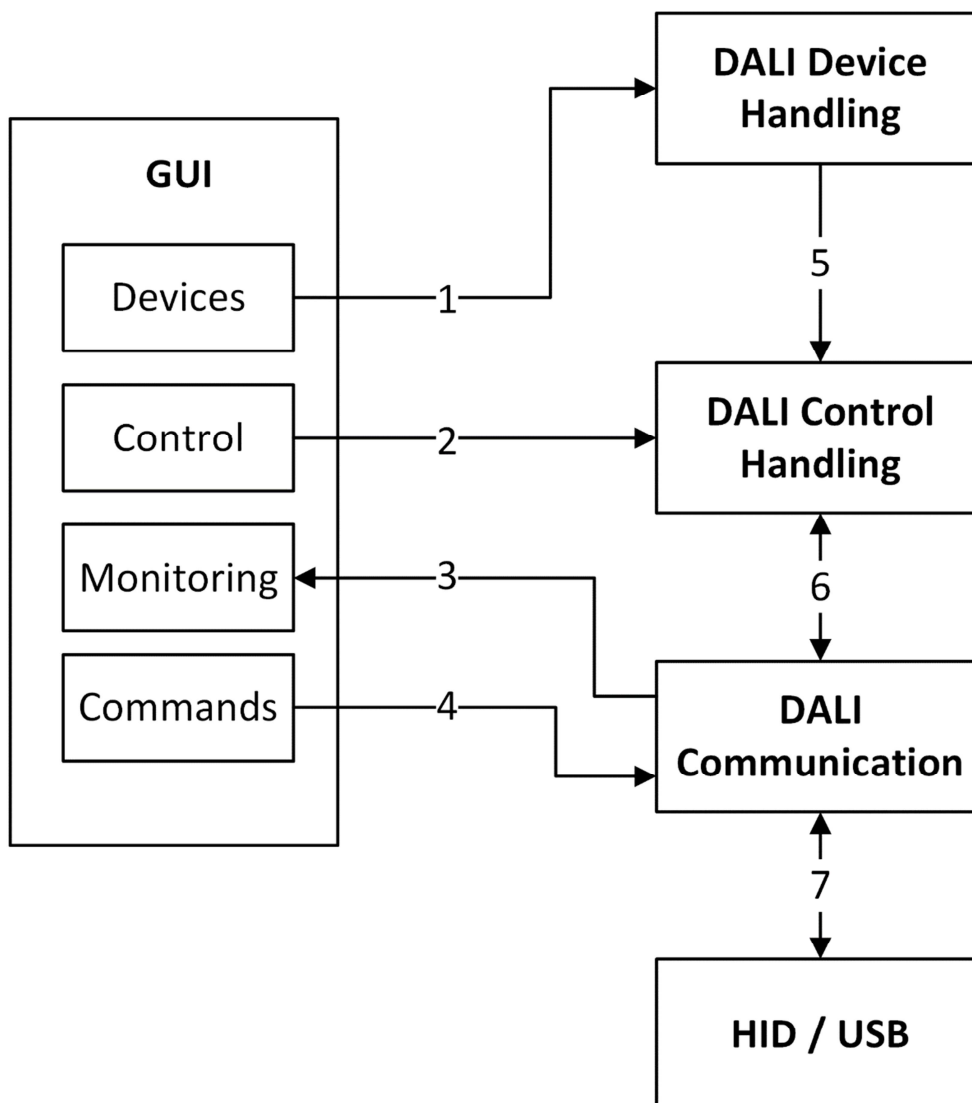
Aplikace je napsána pro operační systém Windows a byla testována na Windows 7. Pro realizaci komunikace přes sběrnici USB využívá aplikace nativních ovladačů operačního systému.

Část aplikace realizující řídicí funkce je psána vícevláknově.

5.3.2 Architektura

Objektově orientovaný charakter jazyka C# dovoluje značně rozmanité možnosti návrhu architektury aplikace, např. pomocí jazyka UML a sofistikovaných nástrojů jako např. Enterprise Architect. Vzhledem ke skutečnosti, že se jednalo o projekt vyvíjený pouze jedním člověkem, a že výchozí koncept aplikace byl jen velice stroze zdokumentován, bylo nakonec od podrobného návrhu architektury upuštěno. Aplikace byla vystavěna na již hotovém výchozím konceptu a rozšířena o další potřebné moduly bez většího předchozího plánování.

Hlavní části aplikace a jejich souvislosti jsou znázorněny na následujícím obrázku:



Obrázek 38 – Základní bloky PC aplikace a jejich souvislosti

Významy jednotlivých bloků na obrázku jsou následující:

- **GUI** – jedná se o blok uživatelského rozhraní. Je rozdělen na další čtyři části
 - **Devices** – obsahuje grafické prvky pro načítání a správu souborů obsahujících informace o jednotlivých DALI Slave zařízeních
 - **Control** – pomocí grafických prvků umožňuje konfigurovat chování celé DALI sběrnice. Umožňuje vytvářet akce a události, pomocí kterých uživatel definuje, jakým způsobem spolu DALI Slave jednotky mohou spolupracovat.
 - **Monitoring** – obsahuje textový výstup o veškerém dění na DALI sběrnici.

- **Commands** – umožňuje uživateli přímo přistupovat na DALI sběrnici manuálním odesláním jednotlivých DALI příkazů
- **DALI Devices Handling** – modul, který realizuje správu dat, ve kterých je uložen popis jednotlivých DALI Slave jednotek. Umožňuje načítání, parsování, a zpracovávání dat z XML souborů.
- **DALI Control Handling** – modul realizující řízení celé DALI sběrnice. Pomocí uživatelsky definovaných pravidel operuje se zařízeními definovanými v modulu DALI Devices Handling.
- **DALI Communication** – modul realizující komunikaci po DALI sběrnici. Sestavuje jednotlivé rámce a odesílá je nižší vrstvě ke zpracování.
- **HID / USB** – realizuje implementaci HID třídy na sběrnici USB. Zpracovává odesílání a přijímání dat na úrovni bytů. Využívá nativní ovladače Windows.

Významy souvislostí mezi jednotlivými bloky jsou tyto:

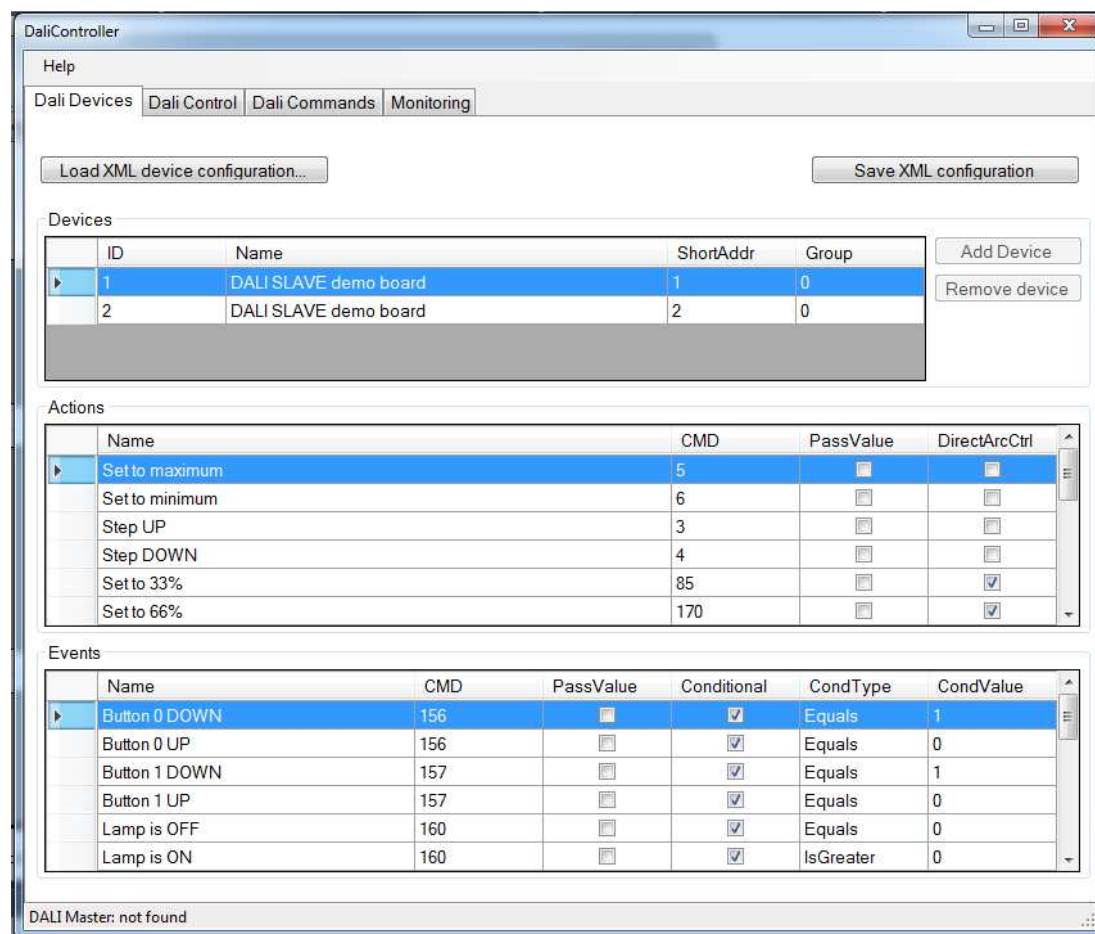
1. Uživatelské interakce s modulem DALI Devices Handling
2. Uživatelské interakce s modulem DALI Control Handling
3. Zpracování všech dat pohybujících se na DALI sběrnici pro účely jejich výpisu do textového výstupu
4. Uživatelské interakce pro přímé odesílání DALI příkazů
5. Načítání dat o DALI Slave jednotkách modulem DALI Control Handling
6. Obousměrná komunikace zajišťující řízení DALI sběrnice
7. Obousměrná komunikace zajišťující výměnu jednotlivých bytů mezi aplikací a DALI Master jednotkou

5.3.3 Grafické rozhraní

Jak už napovídá obrázek v sekci Architektura, je grafické uživatelské rozhraní (angl. zkratka GUI) rozděleno do čtyř hlavních částí. Každá z těchto částí je tvořena samostatnou záložkou na jednom hlavním formuláři – **DaliForm.cs**. Z hlediska použitých knihoven je grafické rozhraní tvořené výhradně knihovnou WinForms.

První dvě záložky, Devices a Control, obsahují kromě běžných grafických prvků, jako jsou např. tlačítka i prvek DataGridView. Jedná se o speciální třídu, která umožňuje zobrazovat datovou tabulku ve formě řádků a sloupců.

Další dvě záložky, Commands a Monitoring, jsou tvořeny běžnými prvky grafického rozhraní.



Obrázek 39 – Ukázka použití DataGridView komponent v PC aplikaci

Aplikace také nabízí několik pomocných formulářů:

- Formulář pro přidávání nového pravidla pro dynamické chování – **RuleAddEditForm.cs**.
- Formulář vypisující chyby – **FormErrorMsg.cs**
- Formulář zobrazující informace o aplikaci – **FormAboutBox.cs**

Jelikož výchozí koncept od společnosti NXP byl realizován v angličtině, byl tento jazyk zachován i v dalších nových prvcích grafického rozhraní. Angličtina byla zachována i ve zdrojovém kódu včetně komentářů.

5.3.4 Správa konfiguračních dat

Z hlediska jejich podstaty lze konfigurační data rozdělit na dvě části: data, která popisují staticky jednotlivé DALI Slave jednotky a data, která popisují dynamické chování celého systému. Data jsou v obou případech uchovávána v souborech XML a každá část je v této kapitole popsána samostatně.

Jelikož samotný jazyk XML neumožňuje pro svoje jednotlivé datové elementy definovat datové typy, každý XML soubor využívá ještě tzv. XML Schema, jenž tento problém řeší. XML Schema mimo to také definuje celou strukturu XML souboru. Aby byl XML soubor považován za validní, musí všechna pravidla stanovená pomocí XML Schema dodržet. XML Schema bývá často realizováno jako doprovodný soubor ve formátu XSD. V našem případě bylo však řešeno tzv. inline způsobem, kdy je struktura XML Schema obsažena přímo v XML souboru na jeho začátku.

Z hlediska implementace je práce s XML řešena pomocí třídy DataSet, kterou platforma .NET nabízí v modulu System.Data. Třída DataSet umožní načíst celý XML soubor včetně XML Schema, sama si jej rozparsovat a vytvořit na základě XML struktury soustavu vzájemně propojených datových tabulek. Na instanci třídy DataSet lze tedy pohlížet jako na jednoduchou formu relační databáze, která se celá nachází v paměti během běhu programu.

Mezi další klíčové vlastnosti třídy DataSet patří zejména:

- Možnost přímého mapování kterékoliv datové tabulky z DataSetu do GUI třídy DataGridView.
- Možnost editace dat v tabulkách pomocí DataGridView nebo jen programově.
- Možnost exportu změněných dat zpět do XML.

Nabízí se otázka, proč data uchovávat v podobě XML souboru, pak je převádět na formu relační databáze a následně opět do XML při ukládání modifikovaných dat. Mohlo by se jevit jednodušší využít rovnou nějaký datový formát umožňující uchovávání informací ve vzájemně provázaných tabulkách. Myšlenka však byla taková, že by data v externím souboru měla být přímo čitelná člověkem a to nezávisle na PC aplikaci, tedy i třeba za pomoci obyčejného textového editoru. Pro tento účel se XML jevil jako velice vhodný nástroj.

5.3.4.1 Data popisující statické vlastnosti DALI Slave jednotky

Všechny DALI Slave jednotky jsou z hlediska PC aplikace popsány v jednom XML souboru. Každé zařízení má definovaný přesný počet parametrů a dále pak libovolný počet tzv. akcí a událostí, které podporuje. V XML souboru je pak jedno zařízení uzavřeno v XML tagu <device>. Z hlediska implementace je zpracování konfiguračních dat popisujících statické vlastnosti DALI Slave jednotky je z velké části řešeno v modulu **DALIDeviceHandling.cs**.

Parametry DALI Slave jednotky

Parametry DALI Slave jednotky jsou definovány jako atributy tagu <device> a jejich seznam je uveden v následující tabulce:

Parametr	Datový typ	Rozsah	Popis
ID	Integer	>0	ID DALI Slave jednotky pro vnitřní účely PC aplikace
Name	String	-	Název DALI Slave jednotky pro vnitřní účely aplikace
ShortAddr	Integer	0-64	Fyzická adresa DALI Slave jednotky
Group	Integer	0-16	Příslušnost do DALI skupiny

Tabulka 7 – Parametry DALI Slave jednotky

Akce podporované DALI Slave jednotkou

Akcí (angl. Action) se rozumí jakákoliv činnost, kterou je DALI Slave jednotka schopna vykonat. Může to být např. rozsvícení či zhasnutí výstupu, jeho nastavení na požadovanou hodnotu anebo třeba přečtení nějaké hodnoty. Zpravidla je akce založená na příkazech DALI Slave jednotky. Uživatel může definovat libovolný počet podporovaných akcí, je však vždy nutné, aby tyto akce byly založené na příkazu, které zařízení podporuje. V XML souboru je seznam akcí uzavřen v tagu <actions>. Každá akce je složená z parametrů uvedených v následující tabulce:

Parametr	Datový typ	Rozsah	Popis
Name	String	-	Název akce pro interní účely aplikace
CMD	Integer	0-255	Kód DALI příkazu spojený s touto akcí
PassValue	Bool	0-1	Určuje, zdali má akce při svém vykonání očekávat vstupní hodnotu
DirectArcCtrl	Bool	0-1	Určuje, zdali se akce přímo nastavuje výstupní úroveň výkonu

Tabulka 8 – Parametry akcí pro DALI Slave jednotku

Události podporované DALI Slave jednotkou

Kromě celé řady akcí může DALI Slave jednotka podporovat i tzv. události (angl. Event). Událostí se rozumí jakýkoliv předem definovaný stav, do kterého se DALI Slave jednotka dostane nebo zaznamená. Událost je rovněž spojena s konkrétním DALI příkazem. Navíc však ještě definuje podmínky, které musí být splněny, aby se výskyt události vyhodnotil jako pozitivní. Typickým příkladem události může být např. stisk tlačítka uživatelem.

Seznam událostí je v XML uzavřen tagem <actions> a stejně jako v případě akcí, i událost je popsána celou řadou parametrů. Jejich výčet je následující:

Parametr	Datový typ	Rozsah	Popis
Name	String	-	Název události pro interní účely aplikace
CMD	Integer	0-255	Kód DALI příkazu spojený s touto událostí
PassValue	Bool	0-1	Určuje, zdali má událost předávat nějakou hodnotu k dalšímu zpracování
Conditional	Bool	0-1	Určuje, zdali je splnění události podmíněno nějakou podmínkou
CondType	Enum	-	Určuje typ podmínky. Může nabývat hodnot rovno, větší než a menší než
CondValue	Integer	0-255	Srovnávací hodnota pro vyhodnocení podmínky události

Tabulka 9 – Parametry událostí pro DALI Slave jednotku

5.3.4.2 Data popisující dynamické chování systému

Stejně jako v případě dat popisující statické vlastnosti DALI Slave jednotek, jsou i data popisující dynamické chování uložena v jednom XML souboru. O správu těchto dat se stará modul **DALIControlHandling.cs**.

Definice pravidel

V XML souboru je definován seznam tzv. pravidel (angl. rules). Každé pravidlo je uzavřeno tagem <rule> a definuje která událost je spojena s kterou akcí a případně jak často má systém ověřovat, zdali k události nedošlo.

Pravidlo se dá chápat jako ucelený předpis toho, jakou předem definovanou akcí má systém zareagovat na danou událost.

Výčet parametrů jednoho pravidla je následující:

Parametr	Datový typ	Rozsah	Popis
Rule_Name	String	-	Název pravidla pro interní účely aplikace
Poll_Interval	Integer	>0	Doba v ms jak často je pravidlo vykonáváno
Event_Addr	Integer	0-64	Adresa zařízení, odkud se očekává událost
Event_Name	String	-	Název události, jak je definována v příslušném XML pro danou Slave jednotku
Action_Name	String	-	Název akce, jak je definována v příslušném XML pro danou Slave jednotku
Action_Addr_Type	Enum	GenericHid. AddrTypes	Určuje o jaký typ adresy akce se jedná (klasická adresa, skupina, broadcast)
Action_Addr	Integer	0-64	Adresa zařízení, pro které je akce určena

Tabulka 10 – Výčet parametrů jednoho pravidla

5.3.5 Řídící funkce

Při zpracování pravidel systém ověřuje, zda jsou příslušné akce a události přítomny a popsány v XML souboru popisu statických vlastností DALI Slave jednotek. Není tedy možné načíst seznam pravidel samostatně, ale vždy musí být doprovázen i druhým zmiňovaným XML souborem.

Před spuštěním vykonávání všech pravidel aplikace sesbírá data z obou XML souborů a vytvoří seznam instancí třídy `RuleEntry`, kde jsou pro každé pravidlo seskupeny jak statické, tak dynamické parametry.

Třída obsahuje navíc ještě proměnnou `executable`, která je datového typu `Bool` a udává, zdali je pravidlo vykonatelné. Toto je zjištěno před spuštěním vykonávání pravidel, kdy aplikace prochází oba XML soubory a kontroluje konzistenci dat. V případě, že narazí na nekonzistentní data – např. pravidlo má v sobě obsažený název události, který v příslušném XML dokumentu není definován – je hodnota `executable` pro toto pravidlo nastavena jako `false`.

Celá podoba třídy jednoho pravidla je následující:

```
class RuleEntry
{
    public Byte eventAddr;
    public String eventName;
    public Byte eventCmd;
    public bool eventConditional;
    public CondTypes condType;
    public Byte condValue;
    public bool eventPassValue;

    public AddrTypes addrType;
    public Byte actionAddr;
    public String actionName;
    public Byte actionCmd;
    public bool actionPassValue;
    public bool directArcCtrl;

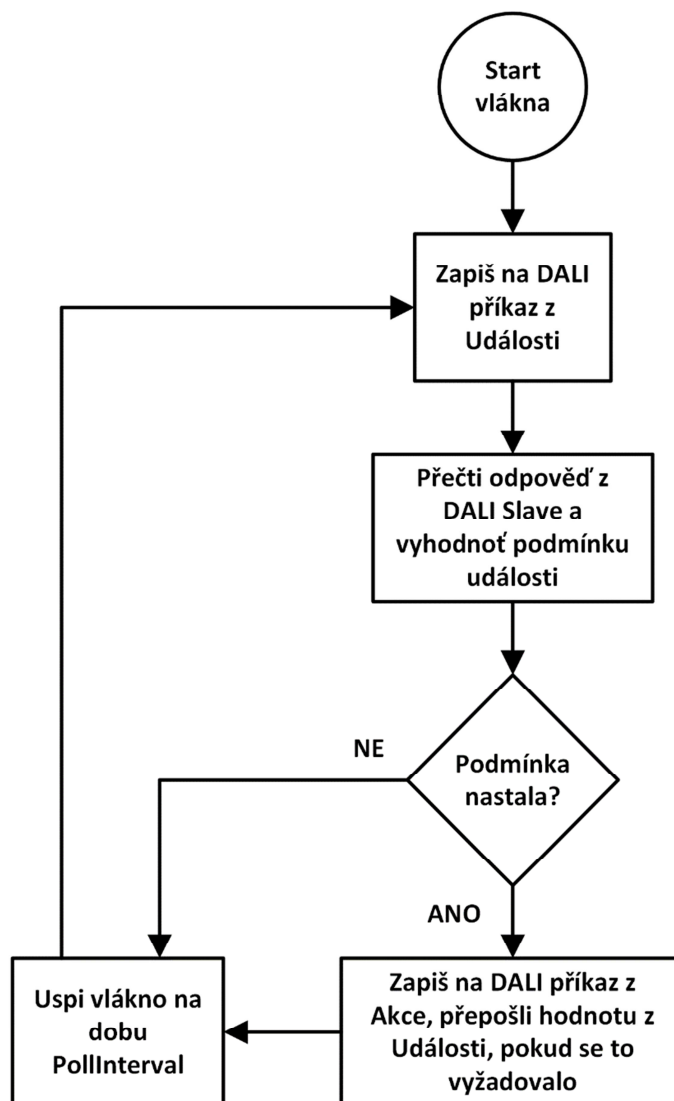
    public int pollInterval;
    public String ruleName;

    public bool executable;
}
```

Vykonávání pravidel

Samotné vykonávání každého pravidla provádí funkce `void ExecuteRule(RuleEntry rule)`, která přijímá jako parametr právě jednu instanci třídy `RuleEntry`. Více pravidel může být vykonáváno současně díky více-vláknovému přístupu.

Na následujícím obrázku je ve zjednodušené formě znázorněný průběh jednoho vlákna pro zpracování jednoho pravidla.



Obrázek 40 – Zpracování jednoho pravidla v PC aplikaci

Z obrázku je patrné, že k uspání vlákna dojde vždy na stejnou dobu podle hodnoty proměnné PollInterval. Je zřejmé, že doba průběhu jednoho vlákna nebude závislá jen na této proměnné, ale i na době průchodu jednotlivými zápisy a čtením z DALI sběrnice. Tyto kroky se mohou časově lišit, např. z důvodu vysoké vytíženosti DALI sběrnice – DALI Slave zařízení může odpovídat déle, případně při vysoké hustotě současně běžících pravidel (vláken) – při pokusech o přístup na DALI sběrnici bude vlákno často blokováno mutexem.

V ideálním případě by bylo dobré měřit čas průchodu celým cyklem a ten pak zohlednit (odečíst) v okamžiku uspávání vlákna. Avšak vzhledem k tomu, že nároky na chod celého

systému v reálném čase nejsou příliš vysoké a také z důvodu zjednodušení vývoje byl tento postup pro zpřesnění doby uspání vlákna vynechán.

Správa vláken

O správu vláken se stará třída `Scheduler`. Třída byla implementována jako statická, může se tedy v aplikaci vyskytovat pouze jednou. Pro účely této aplikace je to však dostačující.

Třída `Scheduler` si samostatně udržuje seznam pravidel (definovaný jako `List<RuleEntry>`), která jsou určena k vykonávání a v případě, že je `Scheduler` spuštěn, pro každé pravidlo z tohoto seznamu vytváří automaticky samostatné vlákno.

Pro účely řízení z jiných částí aplikace implementuje `Scheduler` tyto metody:

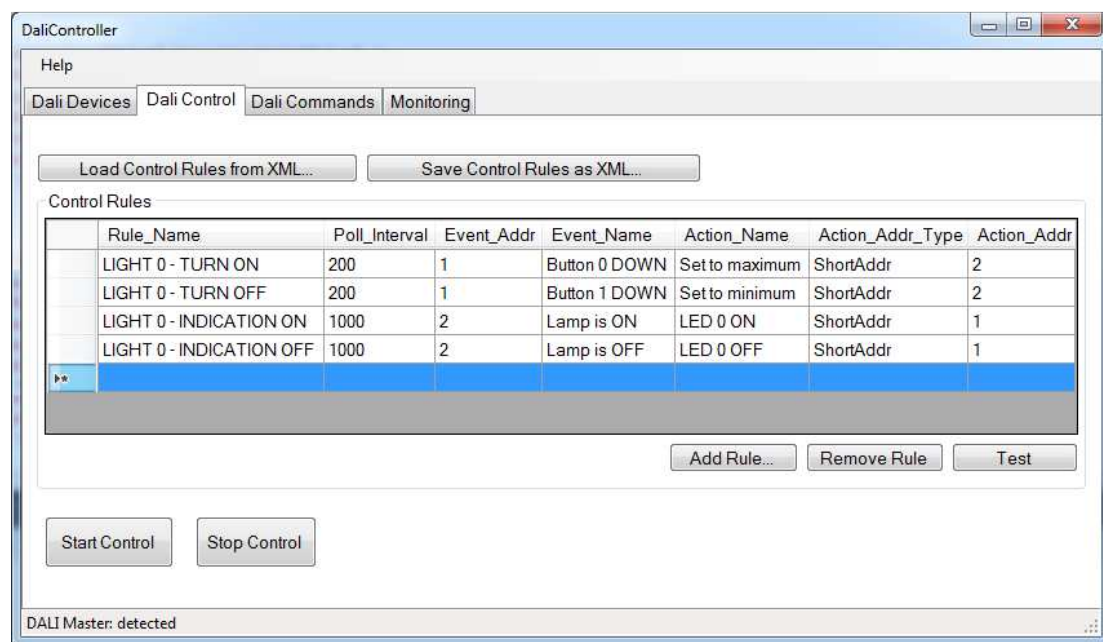
- `void AddRule(RuleEntry rule)` – metoda přijímající jako parametr jeden objekt `RuleEntry` (pravidlo). Toto pravidlo je při zavolání metody přidáno do interního seznamu `Scheduleru`.
- `void ClearRules()` – metoda, jejíž zavolání způsobí vymazání interního seznamu `Scheduleru`.
- `void Run()` – metoda, jenž spustí `Scheduler` (vykonávání pravidel). Po zavolání metody je každému pravidlu z interního seznamu `Scheduleru` přiděleno jedno vlákno a v tomto vláknu je pravidlo vykonáváno.
- `void Stop()` – metoda, jenž ukončí všechna vlákna vytvořená a spuštěná metodou `Run()`.

Příklad použití

Na obrázku níže je vidět jednoduchý příklad řízení funkce ovládání jednoho světla i s indikací. Situace zajišťuje rozsvěcování a zhasínání světla připojeného k modulu s adresou 2 pomocí tlačítek připojených k modulu s adresou 1. Aktuální stav světla je poté indikován LED diodou 0 na modulu s adresou 1.

Situace obsahuje tato čtyři pravidla:

- **LIGHT 0 – TURN ON** – Pravidlo zajišťující rozsvěcování světla na modulu 2. Každých 200ms se dotazuje tlačítka 0 na modulu 1, zdali je stisknuto.
- **LIGHT 0 – TURN OFF** – Pravidlo zajišťující zhasínání světla na modulu 2. Každých 200ms se dotazuje tlačítka 1 na modulu 1, zdali je stisknuto.
- **LIGHT 0 – INDICATION ON** – Pravidlo zajišťující rozsvícení indikační LED 0 na modulu 1. Každých 1000ms se dotazuje světla na modulu 2, zdali svítí.
- **LIGHT 0 – INDICATION OFF** – Pravidlo zajišťující zhasnutí indikační LED 0 na modulu 1. Každých 1000ms se dotazuje světla na modulu 2, zdali svítí.



Obrázek 41 – Ukázka situace řízení jednoho světelného zdroje s indikací

Situace je uložena v XML souboru On-off-example.xml, který je k nalezení v přílohách této diplomové práce.

5.3.6 Komunikace s jednotkou DALI-Master

Třídy a moduly realizující komunikaci s DALI Master jednotkou byly převzaty z původního konceptu aplikace od společnosti NXP. O nižší vrstvu komunikace – obsluhu třídy HID na sběrnici USB se starají moduly **Hid.cs** a **HidDeclarations.cs**

Pro vyhledávání samotného DALI Master zařízení slouží moduly **DeviceManagement.cs** a **DeviceManagementDeclarations.cs**

Nad obsluhou třídy HID je posazen modul v modulu **DALICommunication.cs** realizující komunikaci na úrovni rámců.

Stěžejním prvkem, který umožňuje přístup na DALI sběrnici z vyšších aplikačních vrstev je v tomto modulu tato metoda:

```
bool WriteToDali(  
    Byte daliAddress,  
    Byte daliCommand,  
    bool responseExpected,  
    bool waitAfterSend,  
    out Byte response,  
    out Byte responseStatus  
)
```

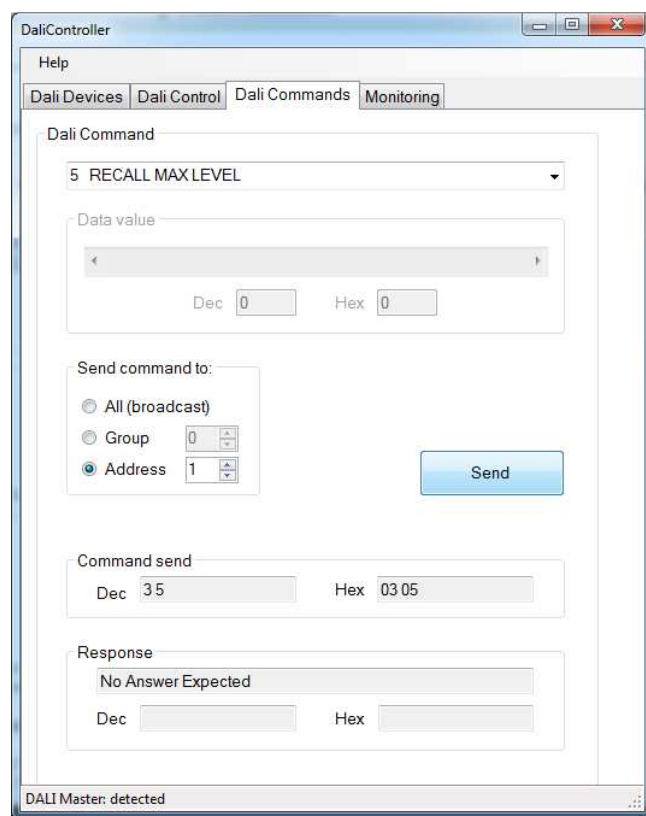
Metoda po zavolání vždy realizuje odeslání jednoho rámce na DALI sběrnici. V závislosti na tom, zdali došla ze sběrnice nějaká odpověď, vrací metoda hodnotu true nebo false. Metoda přijímá řadu parametrů, jejichž význam je následující:

- daliAddress – adresa cílového modulu/skupiny
- daliCommand – DALI příkaz, který se má vykonat
- responseExpected – určuje, zdali se na vyslaný rámec očekává odpověď
- waitAfterSend – určuje, zdali se po odeslání rámce čeká předem definovanou dobu
- response – výstupní proměnná obsahující odpověď na odeslaný rámec
- responseStatus – vypovídá o stavu přijatých dat (OK/no data/no answer)

5.3.7 Přímý přístup na DALI Sběrnici

Využívání řídicích funkcí PC aplikace není jediný způsob, jak přistupovat na DALI sběrnici. Původní koncept aplikace od společnosti NXP nabízel možnost, jak na sběrnici přistupovat přímo. Pomocí jednoduchých grafických prvků si může uživatel navolit jaký příkaz na jakou adresu by měl být po sběrnici DALI odeslán. Tato funkčnost byla v současné podobě PC aplikace zachována. Odesílání dat uživatelem se chová, stejně jako v případě řídicích funkcí, jako samostatné vlákno a může být tedy vykonáváno současně i s dalšími akcemi.

Grafická podoba funkcí pro přímý přístup na sběrnici DALI je zobrazena na obrázku níže.

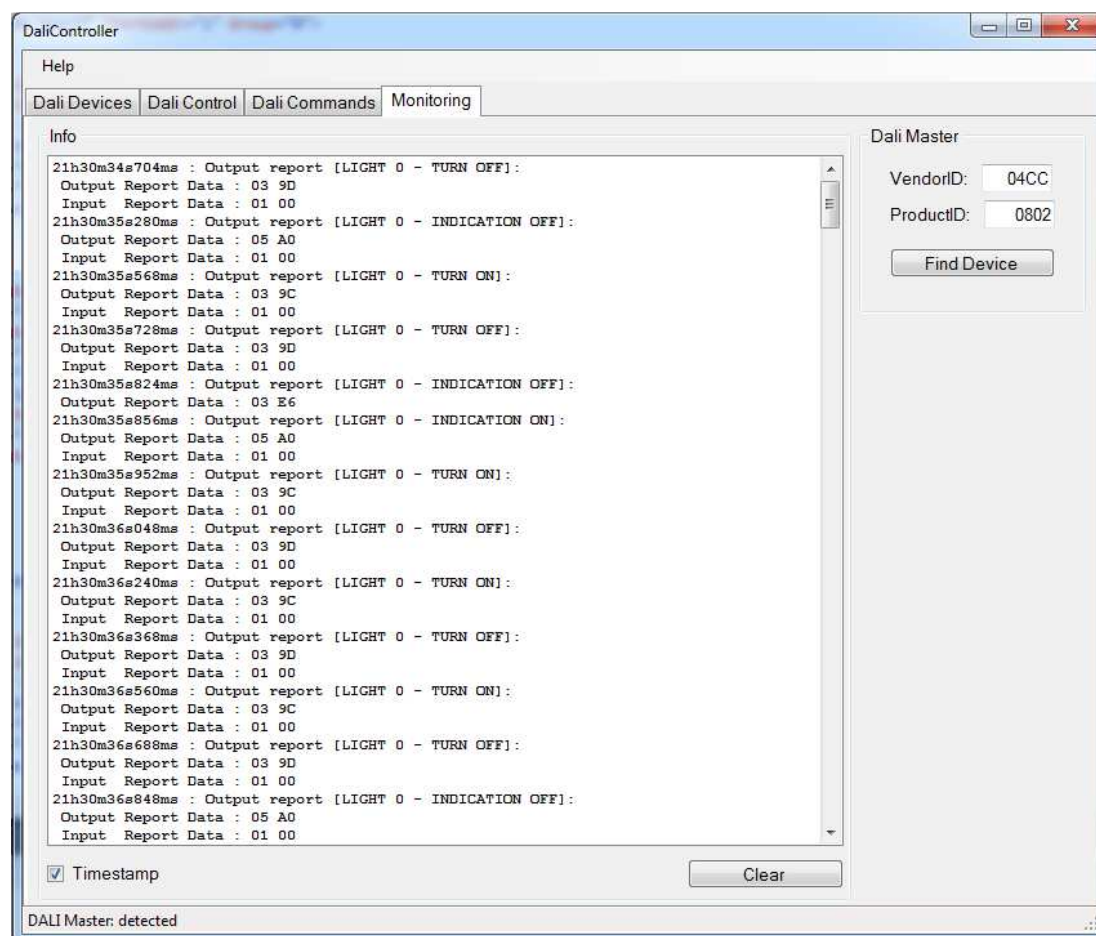


Obrázek 42 – Grafické rozhraní pro přímý přístup na DALI sběrnici

Na rozdíl od řídicích funkcí aplikace, není přímý přístup závislý na definicích jednotek z XML souborů a umožňuje tedy odesílání všech příkazů, které jsou podporovány DALI standardem. Toto je užitečné zejména při odlaďování funkčností neznámých DALI zařízení.

5.3.8 Monitoring

Část aplikace zajišťující monitoring byla rovněž převzata z původního konceptu od společnosti NXP. Monitoring obsahuje jedno velké textové pole, do kterého je vypisována veškerá aktivita na DALI sběrnici. V nadpisu každého záznamu je uveden název pravidla, ze kterého komunikace vzešla. Na obrázku níže je k vidění ukázka záznamu z běhu situace řízení jednoho světelného zdroje. Jedná se o tu samou situaci, která byla již jednou ukázána v předchozích sekcích.



Obrázek 43 – Ukázka záznamu komunikace ze situace řízení jednoho světelného zdroje

ZÁVĚR

V této diplomové práci byly nastíněny některé aspekty používání sběrnice DALI pro řízení světelných soustav a to po stránce teoretické i praktické. V rámci teoretického úvodu do problematiky byly vysvětleny základy komunikace přes tuto sběrnici a také bylo pojednáno o důvodech, proč tuto technologii využívat.

Na základě nejvhodnějšího vybraného konceptu byly zkonstruovány dva typy jednotek umožňující komunikaci přes DALI sběrnici. Funkce těchto zařízení byla demonstrována na testovací fixtuře, která zahrnuje jak samotné jednotky, tak i další příslušenství, které vytvořilo kompletní DALI systém.

V rámci práce byla rovněž navržena a implementována aplikace pro PC, která umožňuje celý systém řídit a konfigurovat.

Autor práce věří, že výsledky jeho snažení mohou posloužit jako solidní základ dalším vývojářům pro realizaci sofistikovanějších a pokročilejších systémů.

CONCLUSION

Some of the aspects of the DALI bus were outlined in this master's thesis both theoretically and practically. As part of theoretical introduction into the topic, basic bus communication principles were explained and also the reasons for utilizing the bus were discussed.

Based on the concept selection, two types of slave units capable of communication over the DALI bus were constructed. The functionality of this devices was demonstrated using a test fixture which besides these devices also utilized other hardware to form a complete DALI system.

In this master's thesis, a PC application allowing the control and configuration of the whole system was also designed and implemented.

The author of this thesis believes that the fruits of his efforts could serve as a solid foundation for other developers to base their more sophisticated and advanced design upon.

SEZNAM POUŽITÉ LITERATURY

1. Semiconductors, NXP. *Application note - DALI slave using the LPC111x*. [PDF] místo neznámé : NXP Semiconductors, 2012. AN11174.
2. Husain, Shaima. *Application note - Digitally Addressable Lighting Interface (DALI) Communication*. [PDF] místo neznámé : Microchip Technology Inc., 2012. AN1465.
3. Philips. *Introduction to DALI*. [PDF] místo neznámé : Philips, 2008.
4. *DALI dimming LED panel light | Theledlight*. [Online] [Citace: 29. Listopad 2014.] <http://www.ledpanellights.com/dali-dimming-led-panel-light.html>.
5. DALI PROFESSIONAL Frequently Asked Questions. *Light is OSRAM | OSRAM*. [Online] [Citace: 29. Listopad 2014.] <http://www.osram.com/media/resource/HIRES/333539/2037986/osram-dali-pro---faq.pdf>. ZPE 2771864 000 00 EN.
6. *DSI Lighting Control*. [Online] Entech(Energy Technologies)Ltd. [Citace: 29. Listopad 2014.] http://www.entech-ltd.com/content/lighting_control_dsi.html.
7. *Lightology | What is 0-10V Dimming?* [Online] Lightology. [Citace: 29. Listopad 2014.] http://www.lightology.com/index.php?module=tools_faq_0_10v_control.
8. *Tridonic GmbH*. [Online] Tridonic. [Citace: 29. Listopad 2014.] <http://www.tridonic.com/com/en/>.
9. *Philips - Česká republika*. [Online] Philips. [Citace: 29. Listopad 2014.] <http://www.philips.cz/>.
10. *Light is OSRAM | OSRAM*. [Online] [Citace: 29. Listopad 2014.] http://www.osram.com/osram_com/.
11. *Helvar*. [Online] [Citace: 29. Listopad 2014.] <http://helvar.com/>.
12. *Routers | Helvar. Helvar*. [Online] [Citace: 29. Listopad 2014.] <http://www.helvar.com/products/routers>.
13. *Helvar*. [Online] [Citace: 29. Listopad 2014.] <http://www.helvar.com/products/dali-keypads>.
14. *Helvar - Lighting control sensors . Helvar*. [Online] [Citace: 29. Listopad 2014.] <http://www.helvar.com/products/lighting-control-sensors>.

15. [Online] Atmel Corporation. [Citace: 29. Listopad 2014.] <http://www.atmel.com/>.
16. *8-bit Atmel Microcontroller with 8K Bytes In-System Programmable Flash*. [PDF] místo neznámé : Atmel Corporation, 2013. 4317K–AVR–03/2013.
17. Alena, Svozilová. *Zlepšování podnikových procesů*. místo neznámé : Grada, 2011. 978-80-247-3938-0.
18. Corporation, Atmel. Atmel AT01244: DALI Slave Reference Design. *Atmel Corporation*. [Online] 01. 07 2013. [Citace: 6. 5 2016.] http://www.atmel.com/images/atmel-42071-dali-slave-reference_design_application-note_at01244.pdf.
19. Optoelectronics, PerkinElmer. *www.perkinelmer.com/opto*. [Online] 2012. [Citace: 6. 5 2016.] <http://www.farnell.com/datasheets/919043.pdf>.
20. Fried, Limor. Adafruit. [Online] [Citace: 14. 5 2016.] https://cdn-learn.adafruit.com/assets/assets/000/010/129/original/APP_PhotocellIntroduction.pdf.
21. HURÁB, Petr. *Huráb - výroba plošných spojů*. [Online] IMFsoft, s.r.o., 2016. [Citace: 6. 5 2016.] <http://www.hurab.com/>.
22. NXP. DALI Development Kit Revision 2. *NXP*. [Online] NXP. [Citace: 6. 5 2016.] <https://www.lpcware.com/content/nxpfile/an11174-and-an11175-lpc11xxlpc13xx-dali-release-20>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

DALI	Digital Addressable Lighting Interface
IEC	International Electrotechnical Commission
AC	Alternating Current
SELV	Separated Extra Low Voltage
CRC	Cyclic Redundancy Check
DSI	Digital Signal Interface
LED	Light Emitting Diode
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
SMD	Surface Mount Device
SPI	Serial Peripheral Interface
MOSI	Master Out Slave In
MISO	Master In Slave Out
FW	Firmware
DPS	Deska plošných spojů
TQFN	Thin Quad-Flat No-leads
PWM	Pulse-width modulation
EEPROM	Electrically Erasable Programmable Read-only Memory
LUT	Look-up table
HID	Human Interface Device
MVC	Model View Controller
XML	Extensible Markup Language
UML	Unified Modeling Language
GUI	Graphical User Interface

SEZNAM PŘÍLOH

- | | |
|-------|---|
| P I | Manuál k řídicí a konfigurační PC-aplikaci |
| P II | Výkresová dokumentace |
| P III | CD se software vč. zdrojových kódů a elektronickou verzí této práce |