

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2018

Roman Sikora



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

RGB MATRIX LED SYSTÉM

RGB MATRIX LED SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Roman Sikora

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lucie Hudcová, Ph.D.

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Roman Sikora

ID: 186184

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

RGB MATRIX LED systém

POKYNY PRO VYPRACOVÁNÍ:

Využijte existující moduly zadní matrix LED lampy a vytvořte pro ně řídicí SW pro platformu Raspberry Pi 3. SW by měl umožnit snadnou implementaci nových animovaných barevných schémat implementujících funkce TAIL, STOP, DIRECTION INDICATOR, FOG a REVERSE, COMING HOME a LEAVING HOME.

Navrhněte 3 barevná schémata pro tyto zadní matrix LED lampy a implementujte je. Navrhněte pro Raspberry Pi ovládací CAN interface a realizujte jej. Vytvořte demonstrační SW pro PC, pomocí kterého bude možné ovládat jednotlivé funkce. Otestujte celek složený z modulů a Raspberry Pi 3 na Conducted emissions.

DOPORUČENÁ LITERATURA:

[1] UPTON, Eben a Gareth HALFACREE. Raspberry Pi: uživatelská příručka. 2., aktualizované vydání. Přeložil Jakub GONER. Brno: Computer Press, 2016. ISBN 978-80-251-4819-8.

Termín zadání: 5.2.2018

Termín odevzdání: 24.5.2018

Vedoucí práce: Ing. Lucie Hudcová, Ph.D.

Konzultant: Ing. Petr Opravil, Hella Autotechnik Nova

prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce se zabývá vývojem řídicího software pro modul zadní RGB LED svítilny umožňující navrhovat a zobrazovat libovolná barevná schémata. Součástí práce je kompletní tvorba software pro platformou Raspberry Pi 3, kterým je svítilna řízená. Tato práce se také věnuje vytvoření uživatelského rozhraní na platformě Windows pro snadný návrh barevných schémat obsahujících typické funkce zadní svítilny jako tail, stop, reverse, fog, direction indicator, coming home a leaving home. Součástí práce je implementace řízení modelu přes automobilovou sběrnici CAN a tvorba řídicího software na platformě Windows. V práci je rovněž celek složený z modelu lampy a Raspberry Pi testován na Conducted emissions.

KLÍČOVÁ SLOVA

RGB, LED, Raspberry Pi 3, tail, stop, reverse, fog, direction indicator, coming home, leaving home, automotive, software, hardware, CAN, Conducted emission

ABSTRACT

The bachelor thesis deals with the development of a control software for the rear RGB LED lamp, which allows to design and display any color schemes. The first part of the work aims at development of complete Raspberry Pi 3 control software. Next thing which the thesis deals with is design of user interface for Windows platform for easy creation of color schemes which includes typical functions for a rear lamps like a tial, stop, reverse, fog, direction indicator, coming home and leaving home. Next part of the work is implementation of the control for the model via CAN bus which is used in automotive technology, and to develop the control software for Windows platform. In the project the whole model with Raspberry Pi is tested for Conducted emissions.

KEYWORDS

RGB, LED, Raspberry Pi 3, tail, stop, reverse, fog, direction indicator, coming home, leaving home, automotive, software, hardware, CAN, Conducted emission

SIKORA, Roman. *RGB MATRIX LED systém*. Brno, 2018, 60 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce: Ing. Lucie Hudcová, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „RGB MATRIX LED systém“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucí práce paní Ing. Lucii Hudcové Ph. D. za trpělivost, aktivitu a cenné rady při realizaci práce, dále bych rád poděkoval konzultantovi práce panu Ing. Petru Opravilovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Rovněž bych rád poděkoval panu Ing. Lukáši Janíkovi za pomoc a cenné rady při realizaci dílčích částí práce.

Brno

.....

podpis autora(-ky)

OBSAH

Úvod	11
1 Teoretický rozbor jednotlivých částí práce	12
1.1 ST LED 524	12
1.2 SPI	13
1.3 CAN	15
1.3.1 Zabezpečení přenosu	17
1.3.2 Signalizace chyb	17
1.3.3 Typy zpráv na sběrnici	18
1.3.4 Can FD	20
1.4 Raspberry Pi 3	20
1.5 Elektromagnetická kompatibilita	22
1.5.1 Conducted emission	23
2 Návrh softwaru pro model zadní svítilny	26
2.1 Raspberry Pi 3	27
2.1.1 BCM2835	27
2.1.2 RGBLED.h	30
2.1.3 Realizace CAN	33
2.1.4 Popis řídicího programu	34
2.2 Uživatelské rozhraní na platformě Windows	39
2.2.1 Grafický vzhled a popis funkcí programu	39
2.2.2 Tvorba animace	41
3 Měření conducted emissions	44
3.1 Měřicí metoda	44
3.2 Výsledky měření	44
4 Závěr	49
Literatura	50
Seznam symbolů, veličin a zkratk	52
Seznam příloh	53
A Fotodokumentace přípravku	54
B Měření conducted emissions	56

SEZNAM OBRÁZKŮ

1.1	Způsob zapojení matice LED diod k obvodu ST LED 524 [2]	12
1.2	Principiální propojení HW prvků SPI rozhraním	14
1.3	Časový diagram SPI rozhraní	15
1.4	Fyzické zapojení Controller Area Network[6]	16
1.5	Časový diagram vodičů CAN[6]	16
1.6	Struktura standardní datové zprávy[6]	18
1.7	Struktura rozšířené datové zprávy[5]	19
1.8	Struktura chybového rámce[5]	20
1.9	Dostupné konektory na jednodeskovém počítači Raspberry Pi 3 model B	21
1.10	Rozložení pinů počítače Raspberry Pi[4]	21
1.11	Základní dělení EMC[9]	22
1.12	Definice úrovně odolnosti a vyzařování[9]	23
1.13	Principiální zapojení napěťové a proudové metody pro měření con- ducted emission[9]	24
1.14	Porovnání P, QP, AV hodnot v časové oblasti[10]	24
2.1	Blokové schéma přípravku	26
2.2	Blokové schéma řídicího programu v RPi	28
2.3	Adresování a rozdělení pattern tabulky s ohledem na využití RGB LED diod	35
2.4	Struktura jednoho bloku dat složeného ze struktur TSVETLO	37
2.5	Struktura binárního souboru s uloženými animacemi	37
2.6	Struktura dat přijímaných přes rozhraní CAN	38
2.7	Ukázka principu aktualizace vykreslované animace	38
2.8	Formulář uživatelského rozhraní pro nastavení rozměrů animace	40
2.9	Vzhled aplikace pro tvorbu animací	40
2.10	Vzhled aplikace pro tvorbu animací	41
2.11	Barvy ohraničení bodů v matici pro jednotlivé funkce	42
2.12	Způsob nastavení barev v aplikaci	42
3.1	Zapojení pracoviště pro měření conducted emissions[13]	44
3.2	Výsledky měření (PK - špičková, AV - střední hodnota, QP - kvazi- špičková) pro šířku pásma mezifrekvenčního filtru 9kHz	46
3.3	Výsledky měření (PK - špičková, AV - střední hodnota, QP - kvazi- špičková) pro šířku pásma mezifrekvenčního filtru 120kHz	47
A.1	Demonstrační model svítilny, je zde naznačeno rozdělení displeje, kdy každou osminu displeje řídí jeden čip STLED524	54
A.2	Demonstrační model RGB svítilny	55

B.1	Umělá síť AMN NNBM 8124 použitá v měření, kdy spektrálním analyzáto- rem je snímaná pouze umělá síť připojená na kladný napájecí vodič.	56
B.2	Spektrální analyzátor Rohde&Schwarz ESR7 řízený počítačem	56
B.3	Uspořádání měřicího pracoviště 1	57
B.4	Uspořádání měřicího pracoviště 2	58
C.1	Stromový výpis obsahu příloženého CD	60

SEZNAM TABULEK

1.1	Tabulka nastavitelných řídicích registrů a popisů jejich funkcí	13
1.2	Tabulka názvů vodičů SPI rozhraní a význam jejich funkce	14
1.3	Tabulka s popisy jednotlivých částí standardní datové zprávy CAN .	18
2.1	Tabulka módů pinů	29
2.2	Tabulka vstupních parametrů <code>bcm2835_spi_chipSelect(uint8_t cs);</code> .	29
3.1	Tabulka limitů rušení pro napěťovou metodu[13]	45

ÚVOD

V posledních letech zažívá vývoj svítlen v automobilovém průmyslu zásadní zlom. Přejít z klasických žárovek na LED (Světlo emitující dioda - Light Emitting Diode) systémy má hned několik důvodů. Jedním z hlavních je bezpochyby úspora energie svítlen tvořených LED diodami. Dalším důvodem je umožnění měnit vyzařování světla v jednotlivých směrech, čímž se dosahuje například maximálního osvětlení prostoru před automobilem, aniž by docházelo k nepříjemnému oslnění řidičů jedoucích v opačném směru, chodců nebo jiných účastníků provozu. Další, bezpochyby žádanou vlastností, je možnost animování jednotlivých částí světla, tedy různé animace při zapnutí, vypnutí, ale i tvorba uvítacích animací apod. Obecně lze říci, že použití LED diod v této oblasti velmi rozšiřuje možnosti designérů.

Tato práce se zabývá programováním nového prototypu zadní svítily pro demonstrační účely tvořené maticí RGB (barevný model červená, zelená, modrá - Red, Green, Blue) LED diod. Potenciál zadní svítily z trojbarevných LED diod spočívá v možnosti vytvoření libovolného tvaru světla bez nutnosti měnit HW například podle požadavků zákazníka, ale také změny tvaru světla za provozu, kupříkladu při změně módu řízení jako „sport“, „economy“ apod. Další výhodou je možnost tvoření libovolných animací při aktivaci jednotlivých funkcí.

Součástí práce je rovněž zhotovení uživatelského rozhraní na platformě Windows pro snadnou implementaci nových animovaných barevných schémat, ale rovněž pro pohodlné ovládání lampy. Model svítily je řízen mikropočítačem Raspberry Pi 3, který sice není možné využít v reálném provozu kvůli nesplnění norem automotive, ale pro demonstrační účely je vyhovující. Komunikace mezi stolním počítačem a RPi (Raspberry Pi) je realizována pomocí CAN (Controller Area Network) sběrnice.

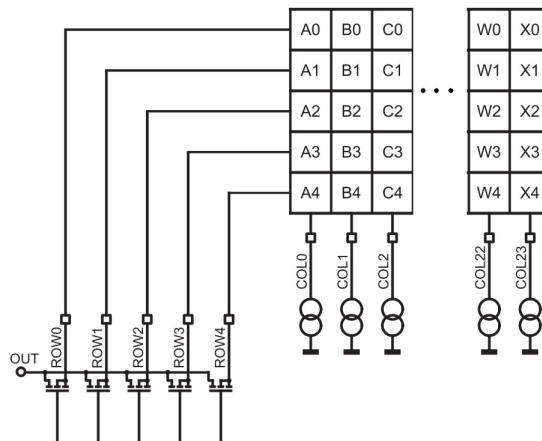
Vzhledem k faktu, že v poslední době velmi narostl počet zařízení, které obsahují elektroniku, je nutné zajistit, aby se tato zařízení neovlivňovala. Z toho důvodu jsou v této práci pro celek, složený z modelu svítily a řídicího počítače, testovány rušivé signály, které se tvoří na napájecím vedení.

1 TEORETICKÝ ROZBOR JEDNOTLIVÝCH ČÁSTÍ PRÁCE

Demonstrační model RGB LED svítily, pro který byl úkol vytvořit programové vybavení, byl zhotoven jako diplomová práce panem Ing. Denisem Hofmanem [1]. Model je tvořen maticí 20x16 RGB LED diod. Celá matice je rozdělena na 8 částí 5x8 diod, kde každá část má svůj řídicí blok a je v podstatě nezávislá na okolí. Ovládání jednotlivých matic je realizováno čipem ST LED 524 [2]. Řízení osmi čipů ST LED je uskutečněno mikropočítačem RPi. Komunikace mezi čipy a RPi je tvořena SPI (seriové periferní rozhraní - serial peripheral interface) rozhraním [1].

1.1 ST LED 524

Jak již bylo zmíněno, každá osmina displeje je ovládaná jedním čipem ST LED 524. Jedná se o prvek vyvinutý společností STMicroelectronics pro ovládání matice LED diod o rozměrech 5 řádků a 24 sloupců. Jak je vidět na *Obr. 1.1*, řádky matice jsou pomocí tranzistorů připojeny ke kladné polaritě, tranzistory zajišťují přepínání mezi řádky v pevně daných časových intervalech. Každý sloupec je připojen k zemi přes proudové zrcadlo, které plní funkci nastavitelného zdroje proudu. Přepínáním mezi řádky a rychlou změnou nastavení proudových zrcadel je dosaženo možnosti ovládní každé diody matice zvlášť.



Obr. 1.1: Způsob zapojení matice LED diod k obvodu ST LED 524 [2]

Obvod obsahuje 23 vnitřních řídicích registrů, z toho je 19 nastavitelných a 4 pouze pro čtení. Dále jsou v obvodu dva soubory registrů pojmenované *pattern1* a *pattern2*. Oba soubory obsahují 240 registrů. Pro každou LED diodu matice je vždy

jeden určující proud diodou v maximálně 255 krocích a druhý definující zpoždění a sklon rozsvícení diody. Následující *Tabulka 1.1* zkráceně popisuje funkci řídicích registrů.

Tab. 1.1: Tabulka nastavitelných řídicích registrů a popisů jejich funkcí

Registr	Funkce
Software control	Registr umožňuje nastavit všechny registry do výchozích hodnot, určuje, zda je displej zapnut nebo vypnut
Display control	Registr určuje, zda se má zobrazit <i>pattern1</i> , <i>pattern2</i> nebo žádný z nich
Clock	Registr umožňuje nastavení výstupního CLKOUT pinu, určuje, zda je pro časování displeje použit externí (z CLKIN pinu) nebo interní oscilátor, povoluje synchronizaci z pinu SYNC, určuje referenci nastavení proudových zrcadel
Column control 1, 2, 3	Registry jsou používány pro povolování sloupců
Row control	Registr je používán pro povolování řádků
Blanking time	Registr určuje dobu zhasínání LED diod
Boost control	Nastavení napěťové úrovně, k jaké se připojují řádky matice
Display visual control	Povolování speciálních funkcí obvodu
PWM control	Nastavení pracovního cyklu PWM (pulzně šířková modulace - pulse width modulation)
Column lock 1,2,3	Blokování sloupců při rolování
Row lock	Blokování řádků při rolování
Scroll control 1, 2, 3	Specifikace rolování (posuvu) displeje
Interrupt enable control	Povolování přerušení generovaných různými zdroji

Komunikace s registry je umožněná přes SPI rozhraní pomocí sady příkazů definovaných výrobcem [2].

1.2 SPI

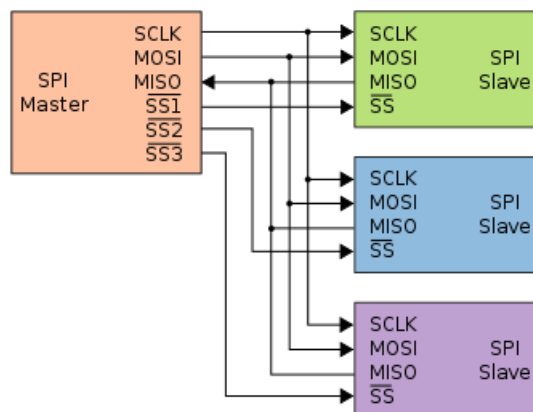
Komunikace mezi jednotlivými čipy, jak již bylo zmíněno, je realizována SPI rozhraním. Plný název zní "Serial Peripheral Interface", tedy sériové periferní rozhraní. Jedná se o sériovou synchronní komunikaci používanou zejména pro mikropočítačovou techniku. SPI rozhraní bylo vyvinuto koncem osmdesátých let 20. století společností Motorola. Architektura sběrnice obsahuje jeden řídicí prvek nazývaný jako

master a několik řízených tzv. slave prvků.

Tab. 1.2: Tabulka názvů vodičů SPI rozhraní a význam jejich funkce

Vodič	Funkce
MOSI	<u>Master Output Slave Input</u> - po tomto vodiči master sériově vysílá data slaveu
MISO	<u>Master Input Slave Output</u> - po tomto vodiči master sériově přijímá data od slaveu
SCLK	<u>Serial Clock</u> - hodinový signál, generuje master
SS	<u>Slave Select</u> - výběr přijímače (slaveu) pro komunikaci, počet těchto vodičů je roven počtu řízených prvků

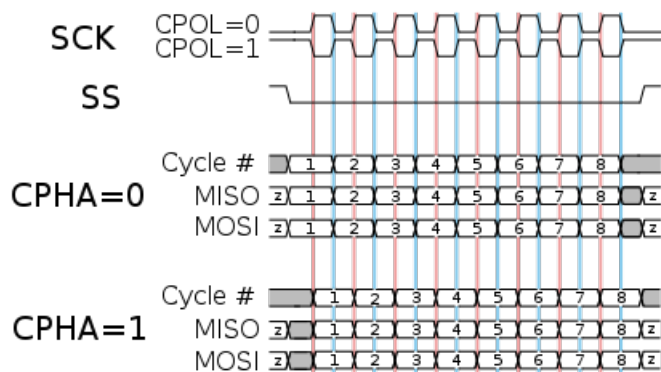
Komunikace mezi zařízeními je plně duplexní, což znamená, že v jeden okamžik mohou téct data z řídicího prvku do řízeného, ale i obráceně. Při zapojení více slaveů je nutné před samotnou komunikací vybrat požadovaný prvek pro komunikaci, a to nejčastěji přivedením nízké úrovně signálu na pin SS (slave select). Princip propojení jednoho prvku master s několika prvky slave je zobrazen na *Obr. 1.2* [3].



Obr. 1.2: Principiální propojení HW prvků SPI rozhraním

Samotná komunikace probíhá v několika krocích. Master vybere slave, se kterým chce komunikovat signálem SS, a následně si master připraví data pro odeslání do výstupního registru. V dalším kroku master nakonfiguruje hodinový signál, který je většinou v jednotkách MHz, podle možností přijímače. Po spuštění hodinového signálu se na příslušné vodiče začnou posouvat data z výstupních registrů, většinou v pořadí od nejvyššího bitu po nejmenší. Po odeslání bytu dat dojde k zastavení hodinového signálu. Po odeslání všech dat proběhne změna úrovně signálu SS (odpojení

od obvodu slave) [4]. Nejčastější je komunikace 8-bitovými slovy. Časový diagram SPI rozhraní je zobrazen na *Obr. 1.3*. Popisek CPOL udává polaritu hodinového signálu, CPHA určuje, na jakou hranu SCLK obvodu reagují. Po odeslání potřebného počtu dat je nutné odpojit slave signálem SS [3].



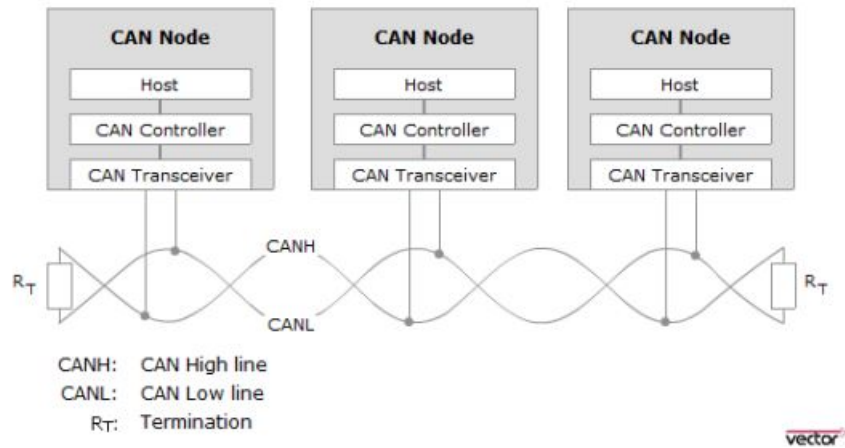
Obr. 1.3: Časový diagram SPI rozhraní

1.3 CAN

V automobilové technice je nutná komunikace mezi mnoha elektronickými řídicími systémy. Vzhledem k potřebám velkého přenosu dat byla společností Bosch vyvinutá sériová sběrnice CAN (Controller Area Network). Z důvodu velkého rozšíření tohoto typu komunikace bude v tomto projektu využít pro komunikaci mezi modelem světla, tedy procesorem RPi a počítačem. Výhodou tohoto systému je možnost ovládní modelu světlů i jinými řídicími obvody umožňujícími CAN komunikaci. Protokol CAN, konkrétně jeho fyzická a linková vrstva, je definována normou ISO11898 [7].

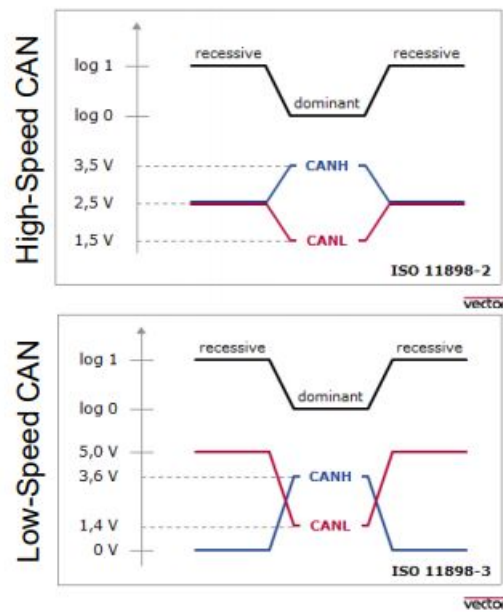
Protokol CAN je typu multi-master, kde každé zařízení připojeno ke sběrnici může být řídicí tak i řízená jednotka. Ke sběrnici může být připojeno až 110 zařízení. Schéma zapojení sběrnice je na *Obr. 1.4*. Tento systém zvyšuje spolehlivost systému, protože pokud nějaký uzel přestane fungovat, zbytek zařízení může komunikovat dál. Komunikace po CAN je velmi dobře zabezpečena proti chybám. Zprávy na sběrnici nejsou adresovány pro konkrétní uzel, ale jsou přijímány všemi zařízeními. Podle obsahu zprávy je poté zpracována příslušným uzlem. V případě kolize více zpráv je přednostně doručena ta s vyšší prioritou [5].

Sběrnice je tvořena dvou vodičovým vedením. Názvy vodičů jsou CAN_L a CAN_H. Pro zamezení odrazů je vedení na obou koncích přizpůsobeno rezistory 120 Ω. Na sběrnici se vyskytují dva stavy. Aktivní (dominantní), který představuje log. 0 a pasivní (recesní), vyjadřující log. 1. V klidovém režimu je vedení v recesním stavu,



Obr. 1.4: Fyzické zapojení Controller Area Network[6]

to znamená, že všechny uzly představují log. 0. Pokud alespoň jeden uzel přejde do dominantního stavu, pak je vedení aktivní. Tyto stavy a napětové úrovně na vodičích jsou zobrazeny na *Obr. 1.5*. Rychlost komunikace dosahuje až 1Mbit/s [6].



Obr. 1.5: Časový diagram vodičů CAN[6]

1.3.1 Zabezpečení přenosu

Jak již bylo zmíněno, CAN protokol disponuje velmi dobrým zabezpečením proti chybám přenosu. Současně zde působí následující mechanismy.

Monitoring: Tato funkce porovnává hodnotu právě vysílaného bitu se stavem na sběrnici, v případě schody obou vysílač pokračuje v odesílání dat. Pokud se hodnoty neshodují a probíhá-li právě řízení přístupu na sběrnici, přerušuje se vysílání a přístup k médiu získá zařízení vysílající zprávu s vyšší prioritou. Pokud je nerovnost hodnot odhalena jinde než při řízení přístupu a v potvrzení přijetí zprávy, vede to ke generování chyby bitu [5].

CRC kód: Cyclic Redundancy Check je slovo o délce 15 bitů na konci vysílané zprávy. Generuje se ze všech do té doby vyslaných bitů. Pokud je detekována chyba CRC libovolným uzlem, je automaticky generována zpráva o chybě CRC [5].

Vkládání bitů (bit stuffing): Pokud se po sběrnici přenáší pět po sobě jdoucích stejných bitů (stejně úrovně), je do zprávy navíc přidán bit opačné úrovně. Tento krok jednak napomáhá k detekci chyb na sběrnici, ale rovněž umožňuje časovou synchronizaci všech zařízení. Pokud dojde ke špatnému vkládání bitů, vygeneruje se příslušná chyba [5].

Kontrola zprávy (message frame check): Zde dochází ke kontrole formátu odesílané zprávy podle specifikace. V případě, že na nějaké pozici je zjištěna nepovolená hodnota, dochází ke generování chyby formátu zprávy [5].

Potvrzení přijetí zprávy (acknowledge): Správné přijetí zprávy musí potvrdit všechny zařízení na sběrnici, včetně těch, které mají zapnuté filtrování a danou zprávu nezpracují. Potvrzení je realizováno změnou bitu v poli ACK z log. 1 vysílané vysílačem na log. 0 [5].

1.3.2 Signalizace chyb

Zařízení připojené na sběrnici počítají množství chyb při příjmu a odesílání. Tím se každé zařízení kontroluje a podle chybovosti se přiděluje do jedné ze tří následujících skupin.

Aktivní (Error Active): Tato skupina uzlů se aktivně podílí na komunikaci, a pokud objeví jednu z chyb výše zmíněných mechanismu zabezpečení, tak vyšlou aktivní příznak chyby (Active Error Flag). Tento příznak je tvořen šesti po sobě jdoucími bity dominantní úrovně, to má za následek porušení přenášené zprávy (poruší se pravidlo vkládání bitů) [5].

Pasivní (Error Passive): Tato skupina se rovněž aktivně podílí na komunikaci, ale uzly vysílají pouze pasivní příznak chyby (Passive Error Flag), který je tvořen šesti bity v recesní úrovni, to nezpůsobí destrukci právě odesílané zprávy [5].

Odpojené (Bus-off): Tyto zařízení jsou odpojeny od sběrnice [5].

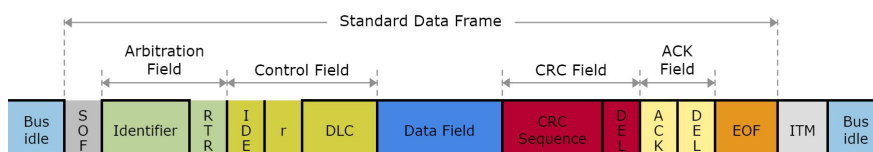
1.3.3 Typy zpráv na sběrnici

Protokol CAN obsahuje několik typů zpráv. Datová zpráva a žádost o data jsou spojené s přenosem dat. Není vždy nutné posílat celou délku datové zprávy, pokud jde o jednoduché informace či pokyny, může se délka slova zkrátit, a tím se zvýší rychlost přenosu protokolu CAN. Další dva typy zpráv jsou chybové [6].

Datová zpráva (Data frame)

Datová zpráva může být realizována ve dvou formátech, *standardním*, který má jedenácti bitový identifikátor a *rozšířeném* s dvaceti devíti bity [5].

Standardní zpráva

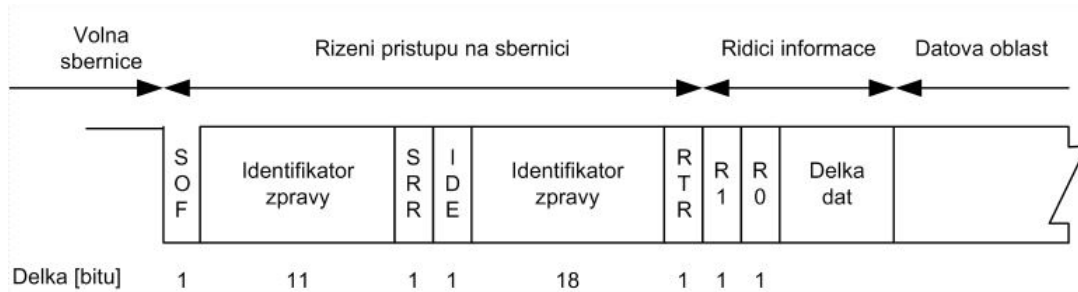


Obr. 1.6: Struktura standardní datové zprávy[6]

Tab. 1.3: Tabulka s popisy jednotlivých částí standardní datové zprávy CAN

Část	Funkce	Bitsy
SOF	Start of Frame - začátek zprávy	1
Identifíer	Identifikátor zprávy	11
RTR	Remote Request - rozlišení typu zprávy, datová zpráva (dominantní), žádost o přístup ke sběrnici (recesní)	1
IDE	Identifikátor Expresion - rozlišení standardního (dominantní) a rozšířeného (recesní) formátu zprávy	1
r	rezervní bit	1
DLC	Data length code - Délka datové zprávy	4
Data Field	Data	0-64
CRC Seq.	Cyklický redundantní součet	15
DEL	Ohraničení CRC (recesní)	1
ACK	Potvrzující bit	1
DEL	Ohraničení ACK (recesní)	1
EOF	End of frame - konec zprávy (všechny bity recesní)	7
ITM	Interframe space - mezera mezi zprávami	3

Rozšířená zpráva obsahuje dvaceti-devíti bitový identifikátor zprávy, který lze rozdělit na dvě části. První část o délce jedenácti bitů, která je totožná s identifikátorem standardního rámce. Další část je přídatná o délce osmnácti bitů. Na Obr. 1.7 je vidět uspořádání záhlaví rozšířeného datového rámce. Bit *SRR* má vždy



Obr. 1.7: Struktura rozšířené datové zprávy[5]

recesní hodnotu pro rozšířený formát zprávy. Tento bit zajišťuje, aby při případné kolizi na sběrnici se standardním rámcem měl priority vždy standardní rámec. Bity *R1* a *R0* jsou rezervní [5].

Žádost o data (Remote frame)

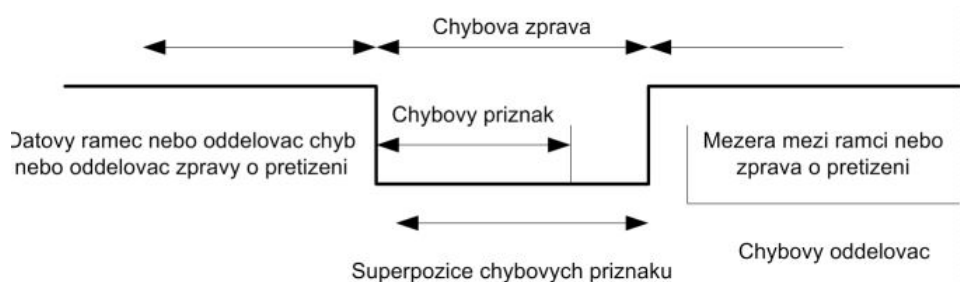
Žádost o data má obdobný formát jako datový rámec, liší se pouze úrovní bitu *RTR* a neobsahuje část *Data Field* [5].

Zpráva o chybě (Error frame)

Zpráva o chybě slouží k signalizaci chyb na sběrnici a je generována ihned poté, co libovolný uzel detekuje chybu. Jak již bylo zmíněno, zpráva může být aktivní (přenášená zpráva je poškozena) nebo pasivní. Délka rámce je od šesti do dvanácti bitů a je dána superpozicí všech chybových příznaků vysílaných jednotlivými uzly [5].

Zpráva o přetížení (Overload frame)

Zpráva o přetížení má podobné složení jako chybový rámec. Rozdíl je, že tuto zprávu lze vysílat i po konci zprávy. Uzel jí vyšle zejména pokud je přetížen a zpracovává předchozí zprávu [5].



Obr. 1.8: Struktura chybového rámce[5]

1.3.4 Can FD

CAN FD (CAN with Flexible Data-Rate) je rozšířená verze protokolů CAN 2.0A a 2.0B (popsaného v kapitole 1.3). Sběrnice byla vyvinutá firmou Bosh v roce 2011 jako odpověď na požadavky přenosu většího množství dat.

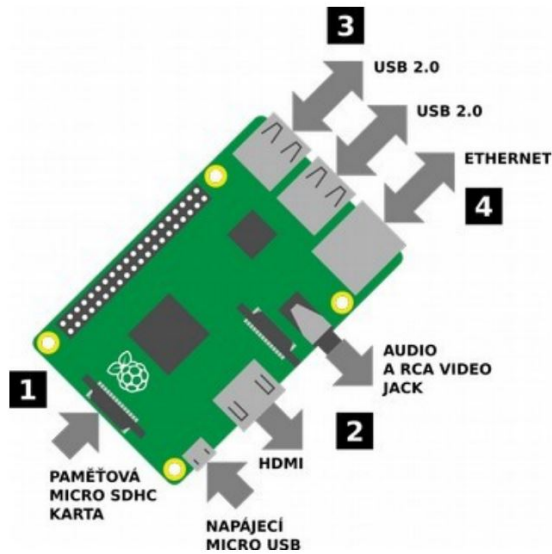
Oproti starší verzi sběrnice umožňuje přenos až 64 bytů v jedné datové zprávě, je zde snížena prodleva mezi instrukcemi a posíláním dat. Dále tento typ snížil počet neodhalených chyb v přenosu díky lepšímu algoritmu CRC. Velkou výhodou je zpětná kompatibilita se starší verzí sběrnice. Rychlost přenosu dat je až 5 Mbit/s [6].

1.4 Raspberry Pi 3

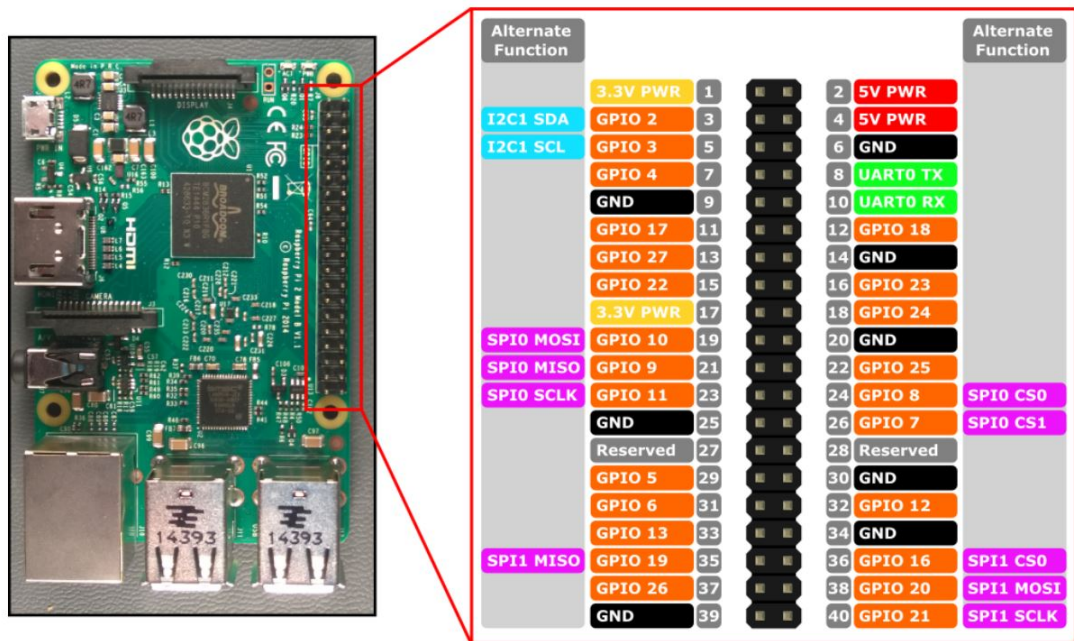
Raspberry Pi je jednodeskový počítač vyvinutý nadací Raspberry Pi Foundantion v roce 2012. Jedná se o možnou náhradu stolního PC s menším výkonem. Software počítače tvoří speciálně upravená distribuce Linuxu Raspbian. Pro používání je nutné připojit pouze monitor, klávesnici a myš. Žádný další HW, jako například pevný disk, grafická karta, není potřeba. Pro napájení je použit síťový adaptér. Dostupné konektory pro připojení periférií jsou zobrazeny na *Obr. 1.9* [8].

Výhodou tohoto počítače je jeho nízká pořizovací cena. Možností využití Raspberry Pi je více. Počítač je možné využít pro běžné potřeby jako je prohlížení internetu, poslouchání hudby, sledování filmů apod. Velikou výhodou Raspberry Pi je, že má vyvedeno 40 vstupně-výstupních pinů, což rozšiřuje možnost jeho využití pro nejrůznější řídicí aplikace. Rozložení pinů a jejich možné funkce je zobrazeno na *Obr. 1.10*. Jak je vidět z obrázku, Raspberry umožňuje komunikaci po několika typech sběrnic I2C, SPI, UART. Výhodou programování aplikace využívající vstupně-výstupních pinů je množství volně dostupných knihoven na internetu. Samotný vývoj aplikace je možné provádět v řadě vývojových prostředí přímo na Raspberry Pi.

Co se týká HW, Raspberry Pi 3 model B je osazen čtyř-jádrovým, 64-bitovým procesorem ARM Cortex-A53 s frekvencí jádra 1,2 GHz a pamětí SDRAM o veli-



Obr. 1.9: Dostupné konektory na jednodeskovém počítači Raspberry Pi 3 model B



Obr. 1.10: Rozložení pinů počítače Raspberry Pi[4]

kosti 1 GB, která je sdílená s grafickým procesorem. Velikost interní paměti je daná použitou MicroSD kartou. Jediný USB 2.0 port, který je vnitřně spojený s ethernet portem, je rozdělen pomocí hubu do čtyřech výstupních USB portů. Raspberry Pi rovněž obsahuje WiFi a bluetooth modul. Rozlišení výstupního videa může být v

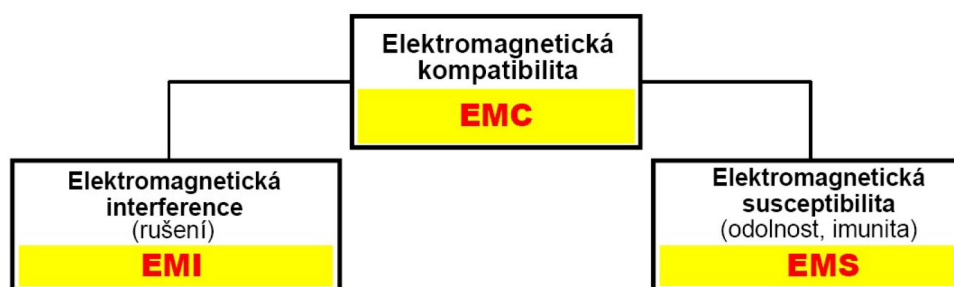
rozmezí od 640x350 do 1920x1200. Rozměry počítače jsou 85,6mm x 56,5mm (bez konektorů), jeho hmotnost je 45g [8].

1.5 Elektromagnetická kompatibilita

Elektromagnetická kompatibilita (EMC) je schopnost elektronických zařízení správně plnit svou funkci i v prostředích, kde se vyskytují zdroje elektromagnetického rušení ať už přírodní nebo umělé. Zároveň tato zařízení nesmí svou elektromagnetickou funkcí nepříznivě ovlivňovat biologická nebo technická zařízení ve svém okolí [9].

Elektromagnetickou kompatibilitu lze dělit podle mnoha různých hledisek. Jejimi hlavními oblastmi jsou EMC biologických systémů, která se zabývá vlivem elektromagnetických signálů na živé organismy a rovněž celkovým elektromagnetickým požadím životního prostředí a EMC technických systémů, která se zabývá vzájemnou koexistencí elektronických zařízení [9].

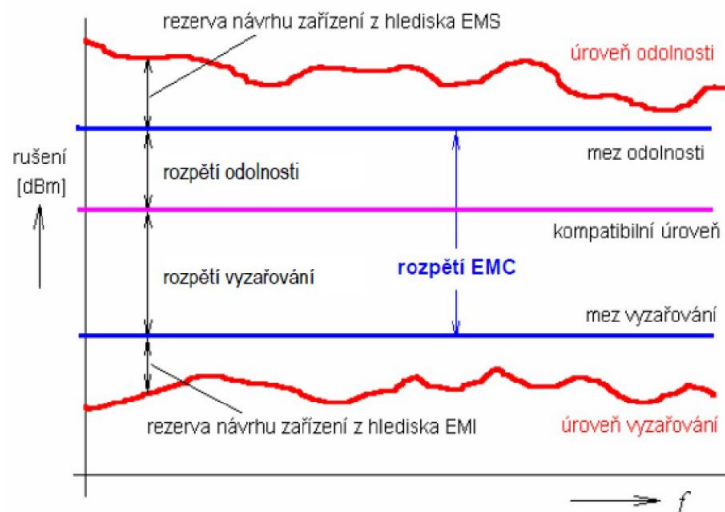
Elektromagnetickou kompatibilitu lze dále dělit na dvě základní skupiny. Elektromagnetickou susceptibilitu (EMS) a interferenci (EMI). Elektromagnetická suscep-



Obr. 1.11: Základní dělení EMC[9]

tibilita (odolnost, citlivost či imunita) vyjadřuje schopnost zařízení pracovat bez poruch či s přesně definovanými odchylkami v prostředí, ve kterém na něj působí vnější elektromagnetické vlivy. Hlavním zaměřením EMS je tedy vytvořit opatření, které zvýší odolnost zařízení proti vnějším elektromagnetickým vlivům. Elektromagnetická interference neboli rušení je naopak část EMI, která vyjadřuje míru interference měřeného zařízení. Toto rušení se prostřednictvím elektromagnetických vazeb šíří do rušených systémů. Hlavním zaměřením EMI je tedy identifikovat zdroj rušení měřením rušivých signálů [9].

Mezi základní pojmy, které jsou definovány Mezinárodním elektrotechnickým slovníkem ČSN IEC 50, patří *Úroveň vyzařování* a *Úroveň odolnosti*. Vysvětlení těchto pojmů je na Obr. 1.12 [9].



Obr. 1.12: Definice úrovní odolnosti a vyzářování[9]

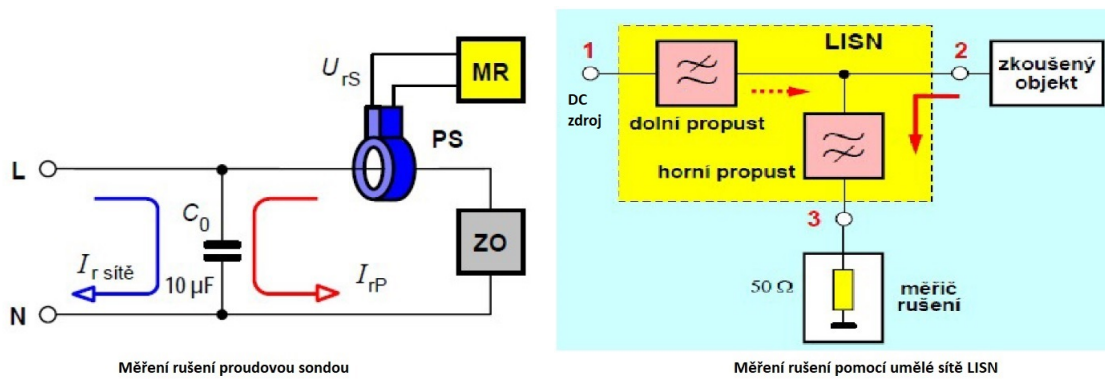
Rušení neboli emise, které je produkováno elektronickým zařízením, se může šířit dvěma způsoby. Pokud je rušivá energie vyzářena do prostoru, tak mluvíme o tzv. rušivém vyzářování (*radiated emission*). Pokud se rušivé signály šíří po vedení, pak se jedná o tzv. *conducted emission*.

1.5.1 Conducted emission

Conducted emission (CE) jsou rušivé signály generované zařízením a šířené po jeho napájecím vedení do jiných komponentů systému nebo do napájecí sítě. Rušivé signály mohou být měřeny napětovou metodou metodou s pomocí sítě LISN nebo proudovou metodou. Tyto signály se měří ve spektru řádově od desítek kHz po desítky MHz. Zdrojem rušení mohou být spínací jevy, přechodové jevy (velká změna zatížení napájecí sítě) a nelinearita zátěže [10].

LISN/AMN

Umělá síť AMN (umělá síť - artificial mains network) označována taky LISN (umělá zátěž vedení - line impedance stabilizing network), slouží k měření rušivého signálu, které vzniká na měřeném zařízení a šíří se po napájecím vedení (conducted emissions). Jak je vidět na *Obr. 1.13*, umělá síť má tři svorky. Svorku 1 pro připojení napájecí sítě, 2 pro připojení zkoušeného objektu a 3 pro připojení měřiče rušení. Vstupní napájecí napětí, z pravidla nízkých kmitočtů, je propouštěno filtrem typu dolní propust do zkoušeného objektu. Vyšší kmitočty, které mohou v síti vznikat,

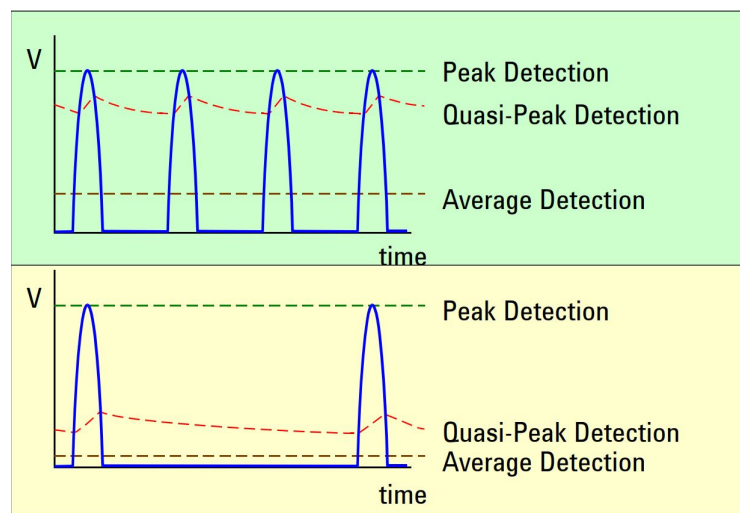


Obr. 1.13: Principiální zapojení napěťové a proudové metody pro měření conducted emission[9]

se do zařízení proto nedostanou. Filtr typu horní propust zajistí, že se napájecí napětí do měřiče rušení nedostane, avšak případné rušivé signály, zpravidla vyšších kmitočtů, filtr propustí. Měřič rušení je většinou tvořen spektrálním analyzátozem [9].

Hodnoty Peak, Quasi-Peak, Average

Při měření CE se měří zpravidla tři základní hodnoty rušivých signálů. Mezi ně patří P (špičková hodnota - peak value), QP (kvazi-špičková hodnota - quasi-peak value), AV (střední hodnota - average value). Grafické porovnání těchto tří hodnot v časové oblasti je na Obr. 1.14.



Obr. 1.14: Porovnání P, QP, AV hodnot v časové oblasti[10]

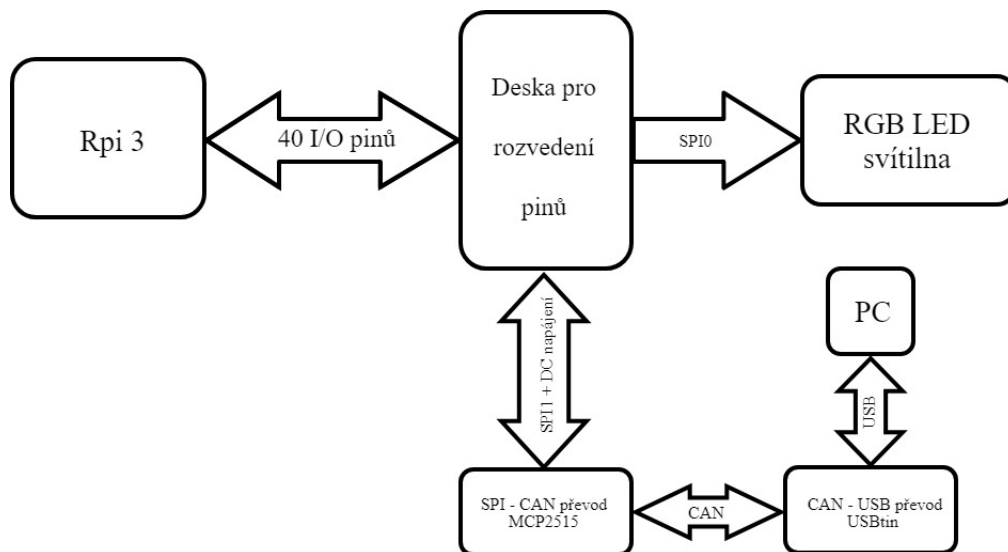
Špičková hodnota je maximální velikost napětí, tedy maximální hodnota obálky na mezifrekvenčním výstupu měřiče rušení. Měření špičkové hodnoty je velmi rychlé a používá se pro přehledové měření v širokém pásmu kmitočtu. Pokud jsou v celém pásmu maximální hodnoty pod limity, není nutno provádět měření ostatních dvou hodnot [10].

Kvazi-špičková hodnota se od špičkové liší v tom, že je úměrná napětově-časové ploše obálky signálu. Je tedy závislá jak na velikosti špiček rušivého napětí tak na jejich opakování. Její nevýhodou je, že měřicí metoda QP je poměrně pomalá, z toho důvodu se aplikuje na užší část spektra tam, kde jsou porušeny limity při měření špičkové hodnoty [10].

Střední hodnota mezifrekvenčního napětí je rovná aritmetické střední hodnotě obálky rušivého signálu [9].

2 NÁVRH SOFTWARE PRO MODEL ZADNÍ SVÍTILNY

Hlavní částí této práce je vytvoření software pro ovládání existujícího modelu zadní RGB lampy. Software se skládá ze dvou částí. První je programové vybavení počítače RPi, který zastává veškeré řízení samotného modelu lampy. Druhou částí je aplikace na platformě Windows, která umožní tvorbu požadovaných barevných schémat. Důležitými požadavky na práci jsou jednoduchost tvorby animací na platformě Windows a zároveň možnost vytvoření jakýchkoliv tvarů základních funkcí svítilny. Software by měl umožňovat tvorbu různých animací pro zapnutí a vypnutí funkce, přepínání mezi sportovním, ekonomickým a normálním režimem, kde sportovní a normální režim mají různé tvary funkcí a ekonomický režim pouze mění intenzitu svícení. Mezi základní funkce zadní svítilny se řadí světla zadní (tail), brzdové (stop), zpáteční (reverse), mlhové (fog) a směrové (direction indicator). Software by měl rovněž umožňovat vytvoření uvítacích funkcí nazvaných "coming home" a "leaving home".



Obr. 2.1: Blokové schéma přípravku

Nejdůležitějším rozhodnutím pro celou práci bylo, jaký zvolit přístup k tvorbě animací. Existuje více způsobů, jak k této problematice přistupovat. Jedním z nich je vektorový přístup, kdy se řídicímu počítači, v tomto případě RPi, zadají potřebné vektory, podle kterých se vypočítá průběh animace. Výhodou tohoto přístupu je jednoduchost pozměnění již hotové animace, možnost přesně vypočítat a nastavit dobu rozsvícení jednotlivých LED diod. Nevýhodou je složitost celého programu a výpočetní náročnost systému. Další možností je bitmapový přístup, kdy si uživatel

jednoduše vytvoří posloupnost barevných bitmap, které se následně cyklicky zobrazují. Výhodou bitmapového přístupu je malá výpočetní náročnost systému, možnost vytvoření zcela libovolných tvarů a animací světla a dále jednoduchost tvorby bitmapových schémat. Nevýhodou však je velká paměťová náročnost, zdouhavá tvorba schémat a obtížné provádění změn v již navržených animacích. Při specifikaci zadání s vedoucím práce bylo rozhodnuto, že se má tento semestrální projekt zaměřit na bitmapový přístup.

2.1 Raspberry Pi 3

Raspberry Pi 3 je základním prvkem celého světla, obsahuje hlavní program, který řídí po SPI rozhraní osm čipů STLED524. Řídící software lampy v RPi je realizován v jazyce C. Jeho úkolem je přijmout a zpracovat data vytvořená v uživatelském rozhraní, kontrolovat požadavky na zapnutí nebo vypnutí jednotlivých funkcí svítilny a nastavit jednotlivé registry řídicích čipů. Blokové schéma programu je na *Obr. 2.2*. Program je založen na principu nekonečné smyčky s vnořenou strukturou switch - case. Na začátku se musí inicializovat potřebné knihovny, nastavit komunikační rozhraní a alokovat potřebnou paměť. Dále se program dostane do zmíněné nekonečné smyčky, na jejíž začátku kontroluje, zda nebyla přijata nějaká řídicí data přes rozhraní CAN. V dalším kroku se případný přijatý signál zpracuje ve switch - case struktuře a vždy na konci smyčky se vyšle do paměti čipů svítilny aktuální hodnoty pro zobrazení.

2.1.1 BCM2835

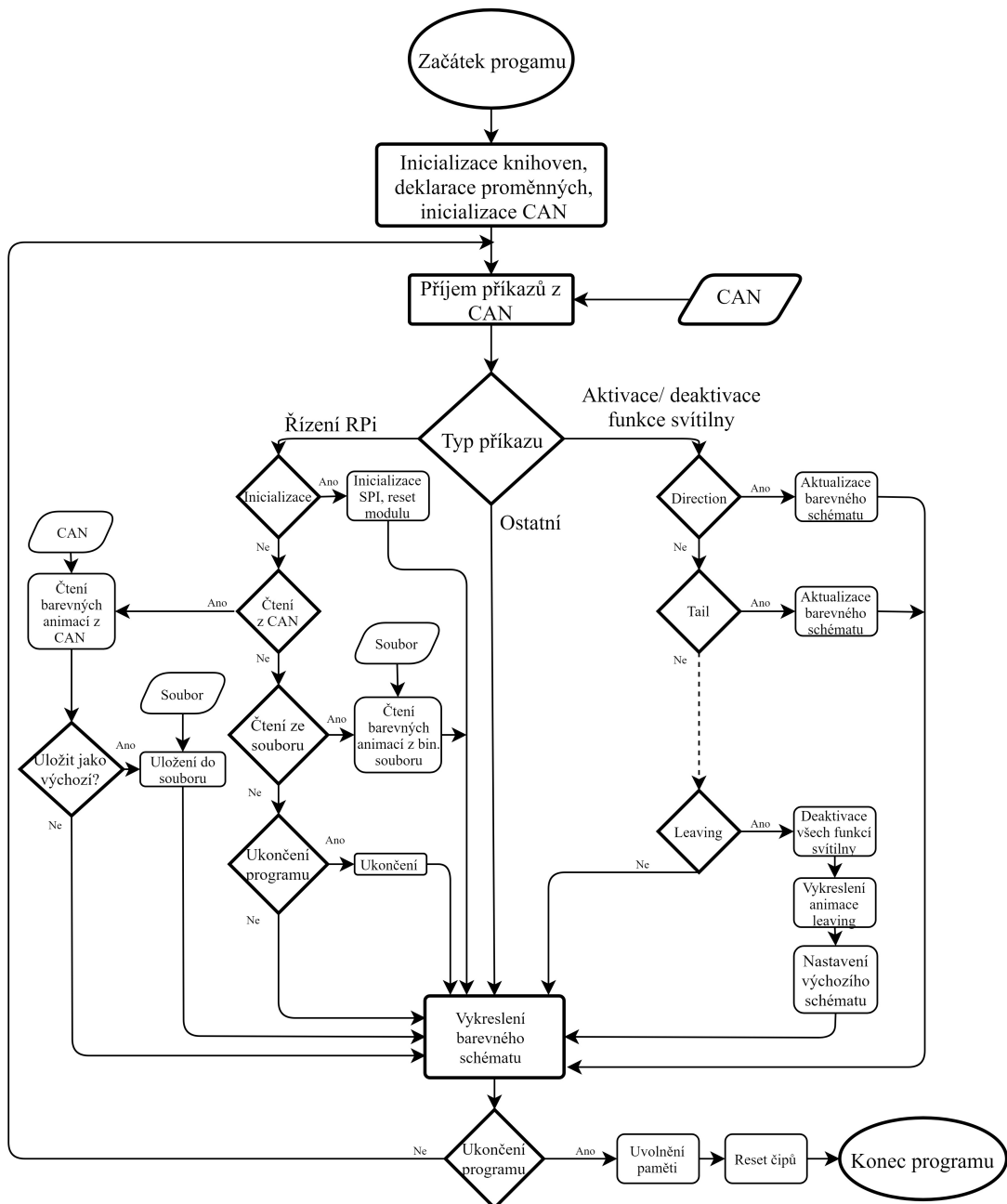
Jedna ze základních knihoven pro využívání GPIO (vstupně-výstupní piny pro všeobecné účely - general purpose input/ output) pinů Raspberry Pi je BCM2835. Jedná se o open-source knihovnu, jejíž autor je Mike McCauley. Knihovna umožňuje číst digitální vstupy, nastavovat digitální výstupy, ale rovněž zprostředkovává používání rozhraní SPI, I2C a využívání časovačů. Přerušení od GPIO pinů knihovna nepodporuje. Největší zastoupení v kódu mají funkce realizující SPI komunikaci. Následuje popis funkcí, které jsou v programu použity.[11]

```
int bcm2835_init();
```

Funkce pro inicializaci knihovny a alokaci potřebné paměti. Jestliže inicializace proběhne úspěšně návratová hodnota bude *1*, v opačném případě *0*;

```
int bcm2835_close();
```

Funkce pro ukončení používání knihovny a uvolnění alokované paměti. V případě úspěšného konce je návratová hodnota *1*, v opačném případě *0*;



Obr. 2.2: Blokové schéma řídicího programu v RPi

```
void bcm2835_gpio_fsel(uint8_t pin, uint8_t mode);
```

Funkce pro nastavení módu GPIO pinu. Funkce nemá návratovou hodnotu. První argument určuje pin, který se má nastavit a druhý parametr stanoví v jakém módu má příslušný pin pracovat. *Tabulka 2.1* zobrazuje možné módy GPIO pinů, jež jsou reprezentovány aliasy *BCM2835_GPIO_FSEL_XX*.

```
int bcm2835_spi_begin();
```

Tab. 2.1: Tabulka módů pinů

XX	Popis
INPT	Vstupní pin
OUTP	Výstupní pin
ALT0-ALT5	Alternativní funkce 0-5

Inicializace SPI komunikace. Funkce nakonfiguruje piny 19 (*MOSI*), 21 (*MISO*), 23 (*CLK*), 24 (*CS0*), 26 (*CS1*) na alternativní funkci *ALT0*. Funkce vrací hodnotu 1, pokud je inicializace úspěšná, jinak vrací 0.

```
void bcm2835_spi_end();
```

Funkce vrací piny SPI rozhraní do výchozího nastavení.

```
void bcm2835_spi_setBitOrder(uint8_t order);
```

Určení posloupnosti bitů při komunikaci SPI. Jako vstupní parametr funkce je možné využít aliasy *BCM2835_SPI_BIT_ORDER_XX*. Pro *XX=LSBFIRST* je posloupnost od nejméně významného bitu po nejvíce významný, pro *XX=MSBFIRST* je to obráceně.

```
void bcm2835_spi_setDataMode(uint8_t mode);
```

Funkce pro nastavení polarity hodinového signálu SPI a jeho fáze.

```
void bcm2835_spi_setClockDivider(uint16_t divider);
```

Funkce pro nastavení hodinového signálu SPI komunikace. Vstupním parametrem je dělitel hodinového signálu jádra procesoru.

```
void bcm2835_spi_chipSelect(uint8_t cs);
```

Touto funkcí určíme, zda pro výběr čipu, se kterým chceme komunikovat, budou použity výchozí piny *CSx* (piny 24, 26), nebo jiné.

Tab. 2.2: Tabulka vstupních parametrů `bcm2835_spi_chipSelect(uint8_t cs)`;

Alias	Popis
BCM2835_SPI_CS0	CS0 na pinu 24
BCM2835_SPI_CS1	CS1 na pinu 26
BCM2835_SPI_CS_NONE	Pro výběr bude použit vlastní, jiný pin

```
void bcm2835_gpio_write(uint8_t pin, uint8_t on);
```

Funkce pro nastavení úrovně na výstupním pinu. První vstupní parametr určuje pin, který se má nastavit. Druhý parametr určí úroveň na pinu, u kterého lze opět použít aliasy *LOW* pro nízkou úroveň a *HIGH* pro vysokou.

```
void bcm2835_delay(unsigned int millis);
```

Zpoždovací funkce, vstupní parametr, je čekací doba v milisekundách.

```
void bcm2835_spi_writenb(char* buf, uint32_t len );
```

Funkce, která vyšle požadovaný počet bytů (parametr *len*) po SPI ze zásobníku. Ukazatel na zásobník je první parametr funkce.

2.1.2 RGBLED.h

Jedná se o hlavičkový soubor obsahující deklarace vytvořených funkcí, definici datového typu a direktivy. Kromě využití několika GPIO pinů jako datových sběrnic je zapotřebí vytvořit 8 signálů *chip select (slave select)* pro výběr obvodu, se kterým chceme komunikovat, jeden signál pro resetování obvodů a jeden synchronizační signál. Pro jednodušší přístup k těmto pinům jsou v hlavičkovém souboru definovány konstanty *CHS1*, *CHS2*, ..., *CHS8*, *RESET*, *SYNC*. Důležitou částí hlavičkového souboru je definování vlastního datového typu *TSVETLO*. Tento typ obsahuje sedm osmibitových proměnných, které následně v programu přesně specifikují jednotlivé LED diody.

```
typedef struct SVETLO
{
    char R;
    char G;
    char B;
    char adresa;
    char display;
    _Bool block;
    unsigned char typ;
}TSVETLO;
```

Proměnné *R*, *G*, *B* specifikují kvantitativní zastoupení jednotlivých barevných složek konkrétní LED diody, proměnná *adresa* odkazuje na přesné umístění v souboru registrů *pattern* v čipu STLED524, kde se mají barevné složky uložit a proměnná *display* specifikuje, na které z osmi části displeje se příslušná LED dioda nachází. Další proměnná *block* je využita při zobrazování daného barevného schématu a určuje, zda se příslušná LED dioda má vykreslit či nikoliv. Poslední datová proměnná *typ* obsahuje informaci o tom, jaké funkce světla (brzda, směrovka, ...) je LED dioda součástí. Následuje popis nejdůležitějších funkcí v RGBLED.h.

```
int SPI_initialisation(void);
```

Funkce inicializuje knihovnu BCM2835, nastaví požadované piny pro výběr čipů jako výstupní, inicializuje SPI komunikaci a nastaví její parametry. Funkce vrací hodnotu *0*, pokud je inicializace úspěšná, jinak vrací *1*;

```
int CAN_init(void);
```

Funkce pro inicializaci CAN komunikace. Funkce vrací deskriptor odkazující na CAN socket, v případě chyby vrací *-1*, pokud se nepodaří otevřít socket a *-2*, pokud se nepodaří přiřadit jméno socketu k adrese.

```
void disp_res(void);
```

Funkce uvede do základního nastavení všechny registry, následně nastaví *boost control register* a *clock register*.

```
void software_control_reg(int num_of_disp, char choice);
```

Touto funkcí je možné uvést registry čipu do původního nastavení a povolit nebo zakázat zobrazování na displeji. Vstupními parametry jsou: *num_of_disp* - číslo displeje, *choice* - výběr (E - povolení zobrazení, D - zakázání zobrazení, A - uvedení všech registrů čipu do původního nastavení, 1 nebo 2 - vynulování *pattern1/pattern2*).

```
void display_visual_control_reg(int num_of_disp, char  
                                registr);
```

Funkce pro povolování speciálních zobrazovacích funkcí čipu. Vstupními parametry jsou: *num_of_disp* - číslo displeje a *registr* - hodnota 8 bitového registru určená podle datového listu.

```
void write_RGBbyte(int num_of_disp, int num_of_pattern,  
                   char adresa, char R, char G, char B,  
                   char SD, char intenzita);
```

Funkce pro nastavení všech tří barev LED diody. Vstupními parametry jsou: *num_of_disp* - číslo displeje, *num_of_pattern* - číslo souboru registrů *pattern 1* nebo *2*, *adresa* - adresa modré barvy dané LED diody v *patternu* (viz. Obr. 2.2), *R*, *G*, *B* - osmi bitové zastoupení tří základních barev, *SD* - specifikace zpoždění a náběhu LED diody při zapnutí a *intenzita* - intenzita svítivosti LED diody v procentech (před samotným zápisem do registrů se každá barevná složka vynásobí procentuální intenzitou).

```
void CHS_H(char CHS) s void CHS_L(char CHS);
```

Nastavení příslušného pinu na úroveň *LOW* nebo *HIGH*. Vstupním parametrem je název nebo číslo pinu.

```
void zero_to_svetlo(TSVETLO** sv);
```

Funkce, která vynuluje strukturu *TSVETLO* a nastaví proměnné adresa a display struktury. Vstupním parametrem je ukazatel na dvourozměrné pole struktur *TSVETLO* o rozměrech 20x16.


```
void povoleni_vykresleni(char typ, _Bool povoleni,
                        TSVETLO** svetlo, int p_r,int p_s);
```

Funkce povolí nebo zakáže vykreslování příslušné funkce svítily v zadaném dvou-
rozměrném poli struktur *TSVETLO*. Vstupními parametry jsou: *typ* - určuje, která
funkce má být upravena, *povoleni* - určuje, zda se vykreslování povolí či nikoliv
(*true/false*), *svetlo* - ukazatel na dvourozměrné pole typu *TSVETLO*, které má být
upraveno, *p_r* - počet řádků pole a *p_s* - počet sloupců pole.

```
void vykresleni(TSVETLO** svetlo, char intenzita, int p_r,
               int p_s);
```

Funkce vykreslí dvourozměrné pole struktur *TSVETLO* na LED svítilnu. Vstupními
parametry jsou: *svetlo* - ukazatel na dvourozměrné pole typu *TSVETLO*, které se má
vykreslit, *intenzita* - intenzita v procentech, s jakou se má daný obrazec vykreslit,
p_r - počet řádků pole a *p_s* - počet sloupců pole.

```
TSVETLO*** make_3D_array(int p_a, int p_r,int p_s);
```

Tato funkce zajišťuje alokaci paměti pro tří-rozměrné pole struktur *TSVETLO*.
Vstupními parametry jsou: *p_a* - počet snímku animace, tedy "hloubku" pole, *p_r* -
počet řádků pole a *p_s* - počet sloupců pole. Funkce vrací ukazatel na trojrozměrné
pole.

```
void prijem_TSVETLO_CAN(int socket_fd_CAN, int p_a, int
                        p_r, int p_s, TSVETLO*** matice);
```

Funkce pro příjem trojrozměrného pole typu *TSVETLO* z rozhraní CAN. Vstupními
parametry funkce jsou: *socket_fd_CAN* - deskriptor nakonfigurovaného rozhraní
CAN, *p_a*, *p_r*, *p_s* - rozměry trojrozměrného přijímaného pole a *matice* - ukazatel
na trojrozměrné pole, kde se mají data uložit.

```
_Bool aktualizace_vykresleni(char typ, _Bool aktualni_stav,
                              TSVETLO*** vykreslovani, TSVETLO*** animace_zapnuti,
                              TSVETLO*** animace_vypnuti, int posledni_vykreslene,
                              int p_a, int p_r,int p_s);
```

Touto funkcí je možné do pole *vykreslovani* nakopírovat vybraná data podle parame-
tru *typ* z pole *animace_zapnuti*, pokud vstupní parametr *aktualni_stav* je *false*, nebo
data z *animace_vypnuti*, pokud je tento parametr *true*. Tato funkce má jednu důle-
žitou vlastnost, snímky (pokud si 3. rozměr pole, tedy "hloubku", představíme jako
počet snímků animace) jedné z výše uvedených animací se nemusí do pole *vykreslo-
vani* kopírovat od začátku, ale i s nějakým ofsetem. Parametr *posledni_vykreslene*
tedy udává, do kterého snímku v poli *vykreslovani* se má začít kopírovat.

2.1.3 Realizace CAN

Jednodeskový počítač Raspberry Pi nepodporuje v základní konfiguraci komunikaci po sběrnici CAN. Mezi jím podporované sběrnice patří I2C, SPI, UART. Z tohoto důvodu je nutné použít převodník, který převede komunikaci po CAN sběrnici na jedno ze zmíněných rozhraní. Jako vhodný pro tuto operaci byl zvolen modul, ve kterém jsou implementovány dva čipy. Prvním je MCP2515, který zajišťuje převod mezi protokoly SPI a CAN, dále je zde čip TJA1050, který realizuje samotnou komunikaci po fyzické sběrnici. Mezi hlavní přednosti modulu patří nízká pořizovací cena, možnost komunikace přes SPI a podpora komunikačního standardu CAN V2.0B. Jeho nevýhodou je nutnost úpravy zapojení v případě použití s počítačem RPi. Napěťová úroveň vstupně - výstupních pinů RPi je 3,3 V, vzhledem k tomu, že čip MCP2515 přímo komunikuje s počítačem po sběrnici SPI, je nezbytné napájet tento čip stejnou napěťovou úrovní, aby se zamezilo poškození pinů. Naopak čip TJA1050 musí být napájen napětím 5 V pro správnou komunikaci po sběrnici CAN.

Další nespornou výhodou tohoto modulu je fakt, že díky jeho velkému využití ve spojení s počítačem RPi vývojáři operačního systému Raspbian zařadili do jádra systému možnost tento modul implementovat jako síťový socket přímo svázaný s rozhraním CAN. Díky tomu je možné po CAN sběrnici komunikovat za použití několika knihoven a jednoduchých funkcí.

Úprava operačního systému Raspbian pro využití modulu MCP2515

Pro používání modulu MCP2515 je nutné udělat v operačním systému několik změn. Na začátku je třeba nainstalovat potřebné nástroje pro využití CAN. Toho lze dosáhnout zadáním následujícího příkazu do terminálu.

```
sudo apt-get install can-utils
```

Dalším krokem je úprava konfiguračního souboru */boot/config.txt*. Jedná se o soubor, který je čten před inicializací procesoru a samotného Linuxu. Dalo by se říct, že je to alternativa tradičního BIOSu v operačních systémech Windows. Do tohoto souboru je nutné připsat několik následujících řádků.

```
dtoverlay=spi=on
dtoverlay=spi1-1cs, cs1_spidev=off, cs2_spidev=off
dtoverlay=mcp2515-can2, oscillator=1600000,
interrupt=25
```

První řádek povoluje používání SPI sběrnice. Druhý řádek specifikuje nastavení SPI, kdy je požadována komunikace pouze s jedním zařízením. Nadbytečné piny pro výběr zařízení pro komunikaci (chip select) jsou deaktivovány. Následující řádek určuje

výchozí přiřazení modulu MCP2515. V tomto případě je to realizováno podle souboru *mcp2515-can2*. Jedná se o speciální soubor typu Device Tree Blob, umístěný v adresáři */boot/overlay*. Tento soubor zajistí přiřazení modulu ke sběrnici SPI 1. Soubor *mcp2515-can2.dtb* není součástí systému a je nutné ho do zmíněného adresáře přidat. Dalším důležitým krokem je nastavení CAN rozhraní. To lze provádět manuálně po každém zapnutí RPi zadáním následujícího příkazu do terminálu (*X* určuje číslo CAN rozhraní, které chceme nastavit, *Y* je požadovaný bitový tok sběrnice).

```
sudo ip link set canX up type can bitrate Y
```

Pro účely tohoto projektu by však manuální zadávání po zapnutí bylo nevhodné, takže je nutné upravit soubor *etc/network/interfaces*. Po vložení následujících řádků na konec souboru se nastavení rozhraní CAN provede automaticky po každém zapnutí.

```
auto canX
iface canX can static
    bitrate Y
```

2.1.4 Popis řídicího programu

Využití speciálních funkcí čipu STLED524 a rozdělení souborů registrů *pattern* podle barev LED diod

Jak již bylo zmíněno, čipy jsou navrženy pro zobrazování obrazců na pole LED diod o rozměrech 5 x 24. Vzhledem k tomu, že jsou v modelu svítily třibarevné LED diody, bude pole diod pouze 5 x 8. Těchto matic je na svítelně 8 a dohromady mají tvořit jeden obrazec.

Z toho plynou různá omezení na použití vnitřních funkcí čipů. Například možnost rolování obrazu není možné použít z důvodu, že nelze vybrat LED diody, které se mají posouvat. Posouvají se vždy všechny, a tak může docházet k nežádoucímu posuvu funkcí, které mají být neměnné, ale jejich část je na rotujícím displeji. Využití PWM regulace pro možnost různých náběžných hran při rozsvícení LED diod se nepodařilo využít. Opět se objevil problém, že při povolení těchto funkcí byla ovlivněna celá část (5x8) displeje, nikoliv pouze požadovaná dioda. Z těchto důvodů jsou vnitřní funkce čipů nevyužity a využívá se pouze možnost nastavení svítivosti LED diody ve 256 krocích v registrech *patternu*.

Důležitou částí kódu je správné adresování jednotlivých diod. Na *Obr. 2.3* je zobrazeno rozdělení jednoho souboru registrů *pattern* s ohledem na využití RGB LED diod. Při použití jednobarevných diod reprezentuje každá pozice jednu diodu, v tomto případě každá RGB LED dioda potřebuje pozice tři.

1 RGB LED dioda

Address	Dimming register	Address	Slope and delay register
0	0-0xFF	1	0-0xE
10	0-0xFF	11	0-0xE
20	0-0xFF	21	0-0xE

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230
2	2	12	22	32	42	52	62	72	82	92	102	112	122	132	142	152	162	172	182	192	202	212	222	232
3	4	14	24	34	44	54	64	74	84	94	104	114	124	134	144	154	164	174	184	194	204	214	224	234
4	6	16	26	36	46	56	66	76	86	96	106	116	126	136	146	156	166	176	186	196	206	216	226	236
5	8	18	28	38	48	58	68	78	88	98	108	118	128	138	148	158	168	178	188	198	208	218	228	238

Obr. 2.3: Adresování a rozdělení pattern tabulky s ohledem na využití RGB LED diod

Inicializace knihoven, deklarace proměnných, inicializace CAN

První částí programu je deklarace vhodných proměnných. Nejdůležitější je vytvořit potřebný počet trojrozměrných polí, kde rozměry polí jsou počet snímků animace, počet sloupců matice a počet řádků matice. Do těchto polí se budou ukládat následující animace: zapnutí a vypnutí v normálním režimu, zapnutí a vypnutí ve sportovním režimu a uvítací "coming" a "leaving". Dále se vytvoří jedno trojrozměrné pole, které se bude vykreslovat a do něhož se budou kopírovat potřebné funkce. Pole nejsou vytvářeny jako statické proměnné, ale v programu jsou deklarovány ukazatele na tato pole, aby byla možnost alokace čím jak nejmenší paměti podle určených rozměrů matice LED diod a počtu kroků (snímků) animace. Názvy polí mají vždy tvar *animace_**.

Další proměnné, které se na začátku kódu deklarují, nesou informaci o tom, zda je aktuálně daná funkce svítily aktivní či nikoliv. Jsou to proměnné, které mohou nabývat pouze hodnot *true*, když je funkce aktivní a *false*, pokud je neaktivní. Název těchto proměnných má tvar **_stav*.

Dále jsou v programu pomocné proměnné nesoucí informaci o rozměrech trojrozměrných polí o tom, který snímek byl vykreslen jako poslední atd. V této části jsou rovněž deklarovány proměnné *rx_frame*, *tx_frame*, které reprezentují rámec dat pro CAN komunikaci. Jedná se o struktury definované v knihovně *can.h*.

V dalším kroku program inicializuje CAN komunikaci pomocí funkce *CAN_init()*. Tato funkce v zásadě otevírá socket CAN speciálními funkcemi z knihoven *can.h*, *raw.h*, *socket.h*. Pokud se funkce vykoná správně, vrátí deskriptor ukazující na socket, ke kterému je přiřazen modul MCP2515. S tímto deskriptorem již můžeme data ze sběrnice CAN číst funkcí *read(...)* a zapisovat funkcí *write(...)* podobě jak

u práce se souborem.

Přijímání příkazů a jejich realizace

Nyní se program nachází v nekonečné smyčce. Na jejím začátku je funkce, která při každém oběhu smyčky kontroluje, zda byla přijata nějaká data prostřednictvím CAN a je nastavená tak, aby neblokovala běh programu, pokud ne.

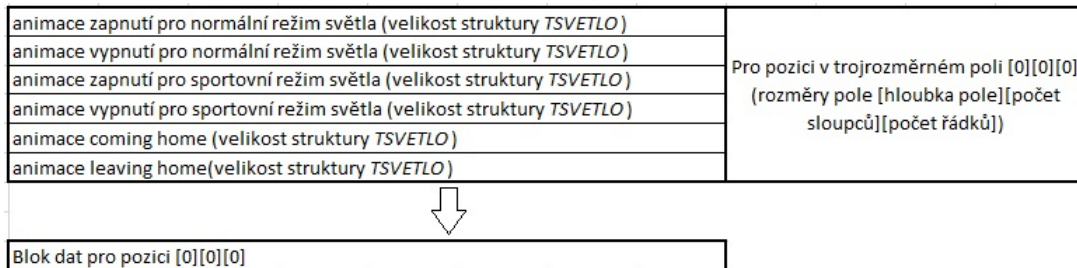
```
nbytes = read (socket_fd, &rx_frame,
               sizeof(struct can_frame));
```

Data, respektive příkazy přijaté v tomto bodě, dělí program na dvě skupiny. Buď se jedná o data pro aktivaci či deaktivaci funkcí svítilny nebo o příkazy, které spouští speciální funkce. Určení, o kterou skupinu jde, je realizováno na základě délky přijatého rámce dat. Dále má v programu každá skupina dat svou vlastní switch - case strukturu, což je možné vidět i na *Obr. 2.2*. V těchto strukturách je pro každý případ v hlavičkovém souboru *RGBled.h* definované klíčové slovo.

Ve skupině, kde se realizují speciální funkce, je několik bloků. Jedním z nich je blok *Inicializace*. Hlavní částí tohoto bloku je inicializace komunikace SPI mezi RPi a svítilnou. Obsah funkce *SPI_initialisation()* zajistí inicializaci knihovny *bcm2835* i sběrnice SPI 0. Nemá vstupní parametry a při použití jiného komunikačního protokolu je nutné změnit obsah samotné funkce. V našem případě je obsahem nastavení potřebných pinů pro výběr čipů *CHS1* až *CHS8* a pinů *RESET* a *SYNC* jako výstupní, nastavení pořadí bitů při komunikaci, které je určeno podle datového listu výrobce čipů STLED524, od nejvíce významného bitu po nejméně významný. Dále se určí polarizaci a fázi hodinového signálu rovněž podle datového listu a zruší se používání předem definovaných pinů pro výběr čipů. Poslední důležitou částí funkce je nastavení děličky hodinového signálu, RPi 3 poskytuje frekvenci 250 MHz, kterou lze následně dělit. Dělička je v tomto případě nastavená na hodnotu 64, což dává výslednou frekvenci o hodnotě 3,9 MHz, a i když výrobce čipů udává maximální frekvenci hodinového signálu SPI 20 MHz, při vyšších frekvencích byla komunikace neúspěšná, což by mohlo být způsobeno nějakým typem rušení mezi RPi a čipy STLED524.

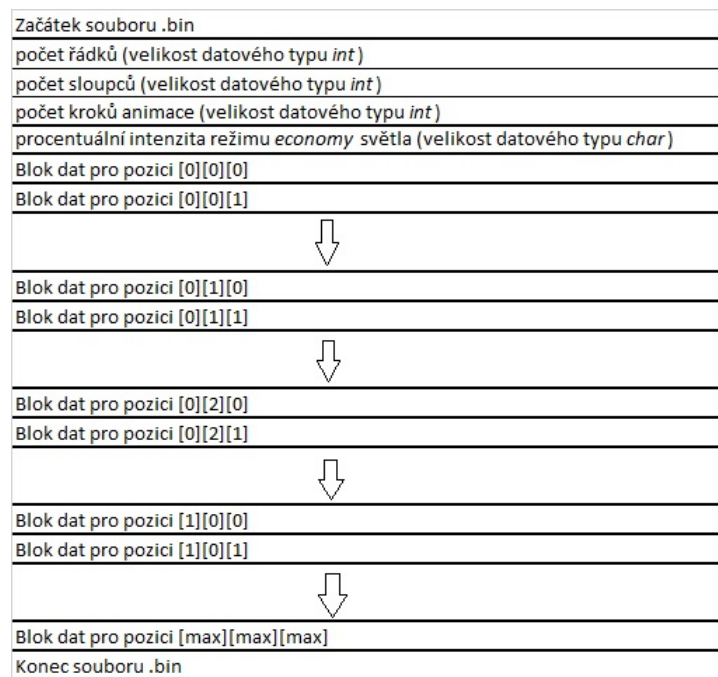
Dále jsou zde dva bloky, které lze použít pro získání barevných schémat pro vykreslování. Jeden z nich je načtení výchozí animace ze souboru. Jedná se o binární soubor, který má pevně danou strukturu viz. *Obr. 2.5*, kde jeden *Blok dat* je složen z jednotlivých struktur *TSVETLO* viz. *Obr. 2.4*. Při procesu načítání ze souboru se jako první přečtou rozměry uložené animace a procentuální intenzita ekonomického režimu světla, následně se alokuje potřebné místo v paměti pomocí funkce *make_3D_array* a až poté dojde k načtení konkrétních animací. Druhý blok umožňuje získat data z PC prostřednictvím CAN. V tomto případě program, po

spuštění příjmu, čeká na odeslání dat z uživatelského rozhraní pro tvorbu animací na počítači. Data začnou přicházet v pevně daném pořadí. Nejdříve, aby se mohla alokovat potřebná paměť, program přijme rozměry matice a ty jsou následovány procentuální intenzitou ekonomického režimu světla. V dalším kroku se alokuje potřebná paměť pro animace a začne se přijímat barevná schémata. Po sběrnici se



Obr. 2.4: Struktura jednoho bloku dat složeného ze struktur *TSVETLO*

posílá každá animace zvlášť. Samotný příjem jednotlivých animací zajišťuje funkce *prijem_TSVETLO_CAN*. Skladba přijímaných dat je naznačena na *Obr. 2.6*. Ob-



Obr. 2.5: Struktura binárního souboru s uloženými animacemi

sah "Blok red" je zastoupení červené barevné složky jednotlivých LED diod v celém trojrozměrném poli animace, data se přijímají po sloupcích. Když se přijmou všechny

sloupce jednoho snímku, posílá se stejným způsobem další. Stejně je tomu u ostatních bloků. Důležitou vlastností funkce pro příjem animací je, že nastaví proměnnou

animace zapnutí v normálním režimu				animace vypnutí v normálním režimu				...
Blok red	Blok green	Blok blue	Blok typ	Blok red	Blok green	Blok blue	Blok typ	...

Obr. 2.6: Struktura dat přijímaných přes rozhraní CAN

block pro každý bod (LED diodu) jednoho snímku animace na hodnotu *true*, pokud se alespoň jedná z barevných složek přijatého snímku liší od hodnoty v minulém snímku, tím se zajistí, že se do řídicích čipů při aktivaci animace pošle vždy jen nejmenší možný počet dat. Každou animaci, kterou se přijme, je možné uložit do binárního souboru jako výchozí.

Další skupina dat, která zajišťuje aktivaci či deaktivaci funkcí svítilny, má svůj vlastní přepínač switch. Přepínání je realizováno podle přijatého příkazu (klíčového slova). V příslušném případě se nejdříve zkontroluje, zda je aktuálně aktivní sportovní režim či nikoliv. To je signalizováno proměnnou *sport_mode_stav*. Podle výsledku se jako vstupní animace funkce *aktualizace_vykresleni* zvolí sportovní nebo normální. Jak již bylo řečeno, parametr *typ* určuje, kterou z funkcí svítilny budeme aktivovat či deaktivovat (0xCC - direction indicator, 0xBB - tail, 0xFF - reverse, 0xEE - fog, 0xDD - stop). Podle vstupního parametru *aktualni_stav* se do pole pro vykreslování nahrají data příslušné animace pro zapnutí nebo vypnutí. Díky vstupnímu parametru *posledni_vykreslene* se nakopírují data do pole, které je vykreslováno tak, aby příslušná animace začala vždy od začátku. Ukázka jednoho případu přepínače je na Obr. 2.7.

```

case DIRECTION:
  if (sport_mode_stav == false) {
    blinkr_stav = aktualizace_vykresleni(0xCC, blinkr_stav, animace_vykreslovani, animace_zapnuti_normal, animace_vypnuti_normal,
                                        posledni_vykreslene, pocet_animaci, pocet_radku, pocet_sloupcu);
  } else {
    blinkr_stav = aktualizace_vykresleni(0xCC, blinkr_stav, animace_vykreslovani, animace_zapnuti_sport, animace_vypnuti_sport,
                                        posledni_vykreslene, pocet_animaci, pocet_radku, pocet_sloupcu);
  }
  tx_frame.data[0] = DIRECTION;
  tx_frame.data[1] = blinkr_stav;
  break;

```

Obr. 2.7: Ukázka principu aktualizace vykreslované animace

Jedním ze zvláštních případů switchu je spuštění ekonomického módu světla. Proměnná *intenzita*, která určuje intenzitu svícení v procentech, je za běžného režimu nastavená na hodnotu *100*. Při aktivaci *economy mode* se její hodnota změní a je nutné aktualizovat všechny spuštěné animace, aby nedocházelo k nežádoucím

předělům, kdy například polovina brzdového světla bude svítit s intenzitou 100% a druhá s nějakou jinou.

Dalším zvláštním případem je aktivace, případná deaktivace sportovního módu v případě, že je nějaká funkce světla zapnutá. Pokud k něčemu takovému dojde, tak se trojrozměrné pole, které se vykresluje, celé vynuluje. Dále se musí překonfigurovat tvary světelných funkcí, tedy na každé pozici v poli se změní proměnná *typ*. Nakonec se do pole pro vykreslování nakopírují nová barevná schémata.

Při aktivaci uvítacích animací se deaktivují všechny funkce svítílny, vymaže se paměť čipů v modelu a následně se vykreslí požadovaná uvítací animace.

Nezávisle na tom, zda program vstoupil do nějaké z výše zmíněných skupin se při každém oběhu hlavní smyčky zobrazí na svítílně jeden snímek pole *animace_vykreslovani* funkcí *vykresleni*. Pořadové číslo snímku, který je aktuálně zobrazen, tedy pozici v "hloubce" trojrozměrného pole, reprezentuje proměnná *posledni_vykreslene*, která se inkrementuje vždy na konci smyčky a může nabývat pouze hodnot 0 až počet snímku animace.

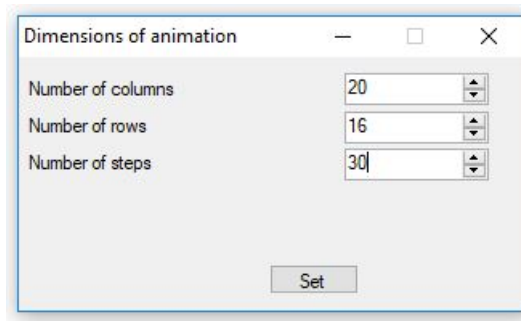
Pokud je na začátku smyčky přijat příkaz pro ukončení programu, tak program vyskočí z nekonečné smyčky. Následně se uvolní pomocí funkce *free*, v průběhu programu, alokovaná paměť. Dále se pomocí funkce *RESET_and_SYNC_L* resetují všechny čipy STLED524 na přípravku. Posledním krokem je vypnutí SPI komunikace a ukončení používání knihovny BCM2835.

2.2 Uživatelské rozhraní na platformě Windows

Pro jednoduchou tvorbu animací a snadné ovládání modulu byl na platformě Windows vytvořen program *RGB_MATRIX*. Jedná se o grafickou formulářovou aplikaci sestavenou ve vývojovém prostředí SharpDevelop. Kód programu je psán v programovacím jazyce C#. Důvodů pro výběr tohoto jazyka bylo několik. Hlavními z nich byly předešlá zkušenost s jeho použitím, podobná syntaxe příkazů s jazykem C, a při využití vhodného prostředí jednoduchá tvorba formulářových aplikací. Jako zmíněné vývojové prostředí byl vybrán SharpDevelop. Jeho výhodou je, že se jedná o program s open-source licencí umožňující vytvářet formulářové aplikace stejným způsobem, jak v prostředí VisualStudio od společnosti Microsoft.

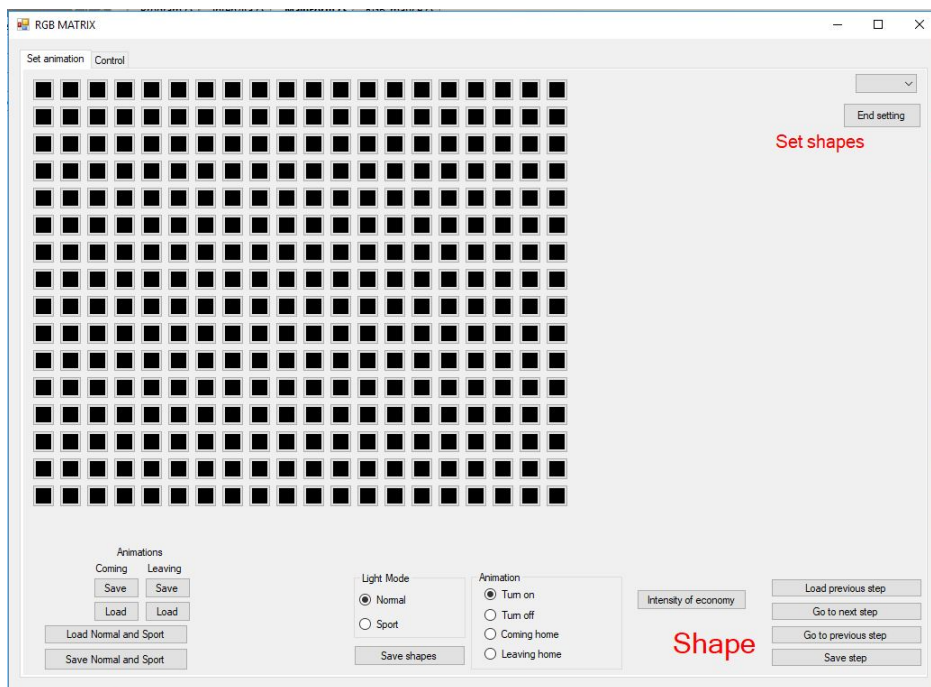
2.2.1 Grafický vzhled a popis funkcí programu

Po spuštění programu se zobrazí formulář (viz. *Obr. 2.8*), ve kterém je nutno zadat požadované rozměry animace, tedy počet řádků a sloupců bitmapy, ale taky množství snímků, které bude výsledná animace obsahovat. Po zadání hodnot se otevře hlavní okno aplikace, na němž je už vykreslená matice požadovaných rozměrů (viz.



Obr. 2.8: Formulář uživatelského rozhraní pro nastavení rozměrů animace

Obr. 2.9). Jak je vidět, formulář obsahuje dvě záložky. První záložka je určena ke tvorbě animací a skládá se z několika částí. Hlavní je matice tlačítek, které reprezentují jednotlivé LED diody na modelu svítily. Dále je zde část pro přepínání mezi

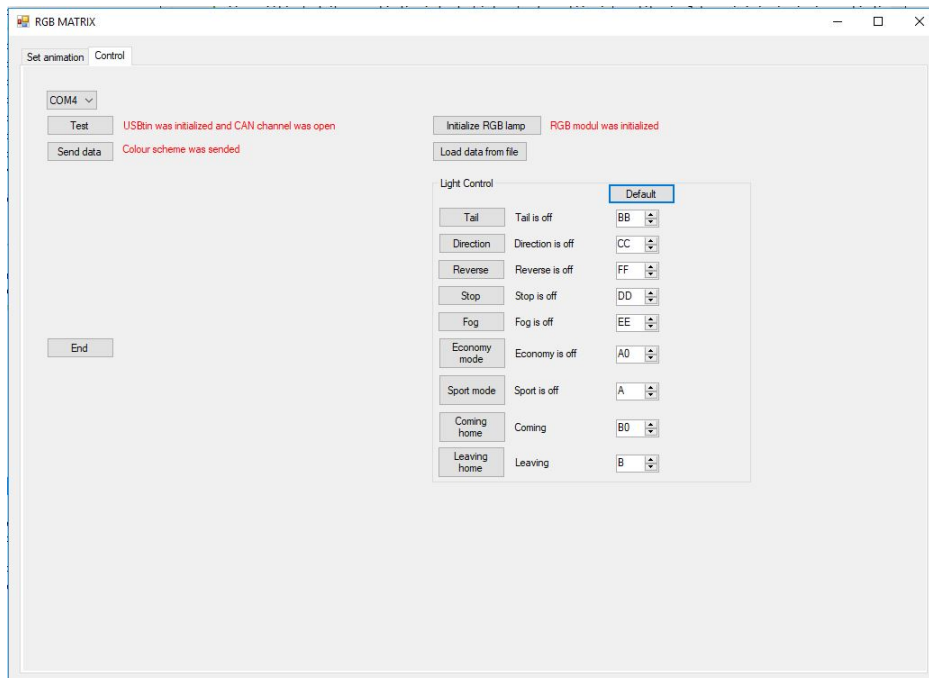


Obr. 2.9: Vzhled aplikace pro tvorbu animací

animacemi a světelnými módy světla, které jsou aktuálně zobrazeny na zmíněné matici. Další části tvoří skupina tlačítek pro pohyb v animaci, tedy přepínání mezi jednotlivými snímky a skupina tlačítek umožňující uložení nebo načtení barevných schémat do souboru.

Druhá záložka je určena pro řízení animace. Jsou zde tlačítka pro inicializaci komunikačního rozhraní a pro odeslání vytvořeného barevného schématu do RPi.

Dále jsou zde tlačítka pro aktivaci či deaktivaci jednotlivých funkcí svítilny s tím, že obsah daného odeslaného příkazu lze nastavit v příslušném numerickém indikátoru. Tyto hodnoty musí odpovídat hodnotám, které jsou definovány pro jednotlivá klíčová slova v RPi. Vzhled záložky je vidět na *Obr. 2.10*.

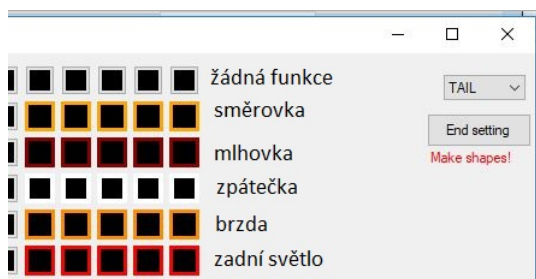


Obr. 2.10: Vzhled aplikace pro tvorbu animací

2.2.2 Tvorba animace

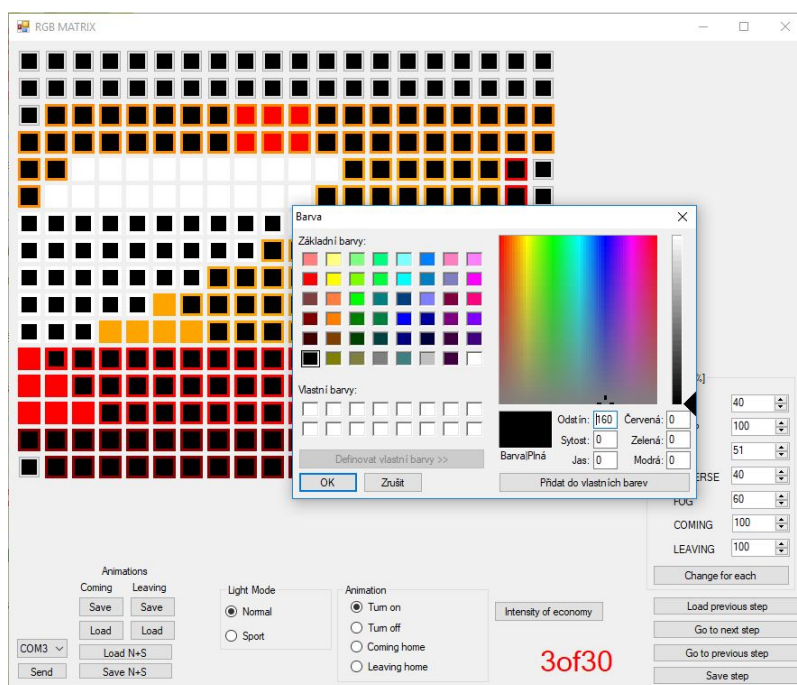
Prvním krokem při tvorbě animace je nastavení tvaru jednotlivých funkcí svítilny. V pravém horním rohu formuláře je rolovací seznam, kde si uživatel vybere funkci, kterou chce tvarovat. Držením pravého tlačítka myši a následným pohybem kurzoru přes body v matici nastavuje požadovaný tvar. Přiřazení bodu k jedné z funkcí signalizuje změna ohraničení daného tlačítka matice (viz. *Obr. 2.11*). Po nastavení tvarů funkcí pro vybraný mód světla je nutné uložení tlačítkem "Save shapes". Následně je třeba stejným způsobem nastavit tvary pro druhý mód světla. Tvary pro zapínací a vypínací animace jednotlivých módů jsou stejné, u uvítacích animací se tvary nenastavují. Když jsou tvary světla nastaveny, je třeba modifikaci ukončit tlačítkem "End setting".

Nyní je možné přistoupit k nastavování samotných barev. Kliknutím na bod v matici se zobrazí paleta barev, kde si může uživatel vybrat ze základních předdefinovaných barev nebo si může definovat svou vlastní barvu nastavením hodnot,



Obr. 2.11: Barvy ohraničení bodů v matici pro jednotlivé funkce

kteří reprezentuje zastoupení červené, zelené a modré složky ve výsledné barvě (viz. Obr. 2.12). Po výběru je možné zmáčknout levé tlačítko myši a přetažením kurzoru nastavit stejnou barvu jiným bodům matice. Když je modifikace barev snímku



Obr. 2.12: Způsob nastavení barev v aplikaci

ukončena, je třeba výsledný krok uložit tlačítkem "Save step". Následně se pomocí tlačítek "Go to next step" a "Go to previous step" může uživatel pohybovat mezi jednotlivými snímky animace a výše zmíněným postupem nastavovat barvy. Vlastností, která urychluje tvorbu animace, je možnost načtení barevného nastavení předešlého snímku pomocí tlačítka "Load previous step". Při tvorbě barevného schématu vypínací animace lze předpokládat, že složení barev prvního snímku sekvence bude stejné, jak u posledního kroku animace pro zapnutí. Z toho důvodu se při tvorbě

sekvence pro vypnutí zobrazí tlačítko "Load last", kterým lze tento krok vyřešit.

Dalším zjednodušením je možnost měnit procentuální intenzitu jednotlivých funkcí. Jak již bylo zmíněno, je možné, po výběru barvy, nastavovat body pouhým přetažením kurzoru, pokud toho uživatel využije a nastaví více funkcím stejnou barvu, může tímto způsobem změnit intenzitu, s jakou bude daná LED dioda svítit. V případě, že jsou všechny animace hotovy, může uživatel přistoupit k odeslání dat nebo vytvořené animace uložit do souboru.

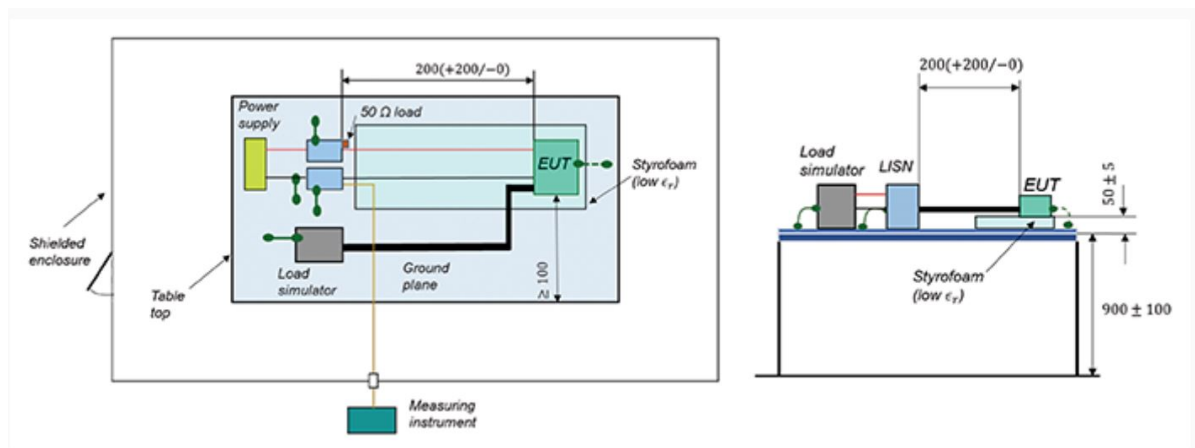
Vzhledem k tomu, že lze animace ukládat, je zde samozřejmě možnost je i načíst. Podmínkou pro tento krok je, aby rozměry uloženého barevného schématu byly stejné jak ty, které uživatel zadá při spuštění programu. Tvary světelných funkcí se automaticky po načtení souboru aktualizují. Pro možnost využití stejných uvítacích sekvencí s více animacemi obsahujícími světelné funkce je jejich ukládání rozděleno do jiných souborů.

3 MĚŘENÍ CONDUCTED EMISSIONS

3.1 Měřící metoda

Jedním z bodů zadání bakalářské práce bylo měření rušivých napětí na napájecím vedení. Měření probíhalo v laboratořích společnosti HELLA AUTOTECHNIK NOVA S.R.O. Měření bylo prováděno podle automobilového standardu CISPR 25 (International Special Committee on Radio Interference), ten určuje limity emisí jak vyzářených do prostoru, tak šířících se po napájecím vedení, ale rovněž popisuje měřící metody.

Zapojení pracoviště popisuje *Obr. 3.1*. Každý napájecí vodič RGB lampy je při-



Obr. 3.1: Zapojení pracoviště pro měření conducted emissions[13]

pojen přes AMN(LISN) NNBM8124 ke zdroji. Jedná se o umělou síť s impedanční charakteristikou realizovanou podle normy CISPR 25, umožňující měření rušení na jednom vodiči ve frekvenčním rozsahu 100kHz až 150MHz. Jako měřič rušivého signálu je použit spektrální analyzátor R&S ESR7.

Standard CISPR 25 určuje frekvenční rozsah měření od 100kHz do 108MHz. Měřená zařízení jsou touto normou rozdělena na pět tříd. Pro každou třídu platí jiné limitní hodnoty rušení na jednotlivých frekvencích. Pro napětovou metodu zobrazuje limitní hodnoty *Tabulka 3.1*.

3.2 Výsledky měření

Měřeny byly špičkové, střední a kvazi-špičkové hodnoty. Měření bylo provedeno pro dvě šířky pásma mezifrekvenčního filtru 9kHz a 120kHz. S první hodnotou bylo proměřeno celé pásmo, pro které je standard CISPR 25 definován, tedy 100kHz až

Tab. 3.1: Tabulka limitů rušení pro napěťovou metodu[13]

Service /Band	Frequency MHz	Levels [dBuV]									
		Class 1		Class 2		Class 3		Class 4		Class 5	
		PK	QP	PK	QP	PK	QP	PK	QP	PK	QP
LW	0,15 - 0,3	110	97	100	87	90	77	80	67	70	57
MW	0,53 - 1,8	86	73	78	65	70	57	62	49	54	41
SW	5,9 - 6,2	77	64	71	58	65	52	59	46	53	40
FM	76 - 108	62	49	56	43	50	37	44	31	38	25
TV band 1	41 - 88	58	-	52	-	46	-	40	-	34	-
CB	26 - 28	68	55	62	49	56	43	50	37	44	31
VHF	30 - 54	68	55	62	49	56	43	50	37	44	31
VHF	68 - 87	62	49	56	43	50	37	44	31	38	25

Service /Band	Frequency MHz	Levels [dBuV]				
		Class 1	Class 2	Class 3	Class 4	Class 5
		AWG	AWG	AWG	AWG	AWG
LW	0,15 - 0,3	90	80	70	60	50
MW	0,53 - 1,8	66	58	50	42	34
SW	5,9 - 6,2	57	51	45	39	33
FM	76 - 108	42	36	30	24	18
TV band 1	41 - 88	48	42	36	30	24
CB	26 - 28	48	42	36	30	24
VHF	30 - 54	48	42	36	30	24
VHF	68 - 87	42	36	30	24	18

108MHz, s šířkou pásma 120kHz bylo proměřeno pásmo od 30MHz do 108MHz. Výsledky jsou zobrazeny v grafech, kde jsou rovněž vyznačeny limitní hodnoty pro pátou třídu zařízení podle zmíněného standardu. Velikost rušení je zde zobrazena v jednotkách $dB\mu V$. Jedná se o logaritmickou jednotku, kterou lze vypočítat následujícím způsobem.

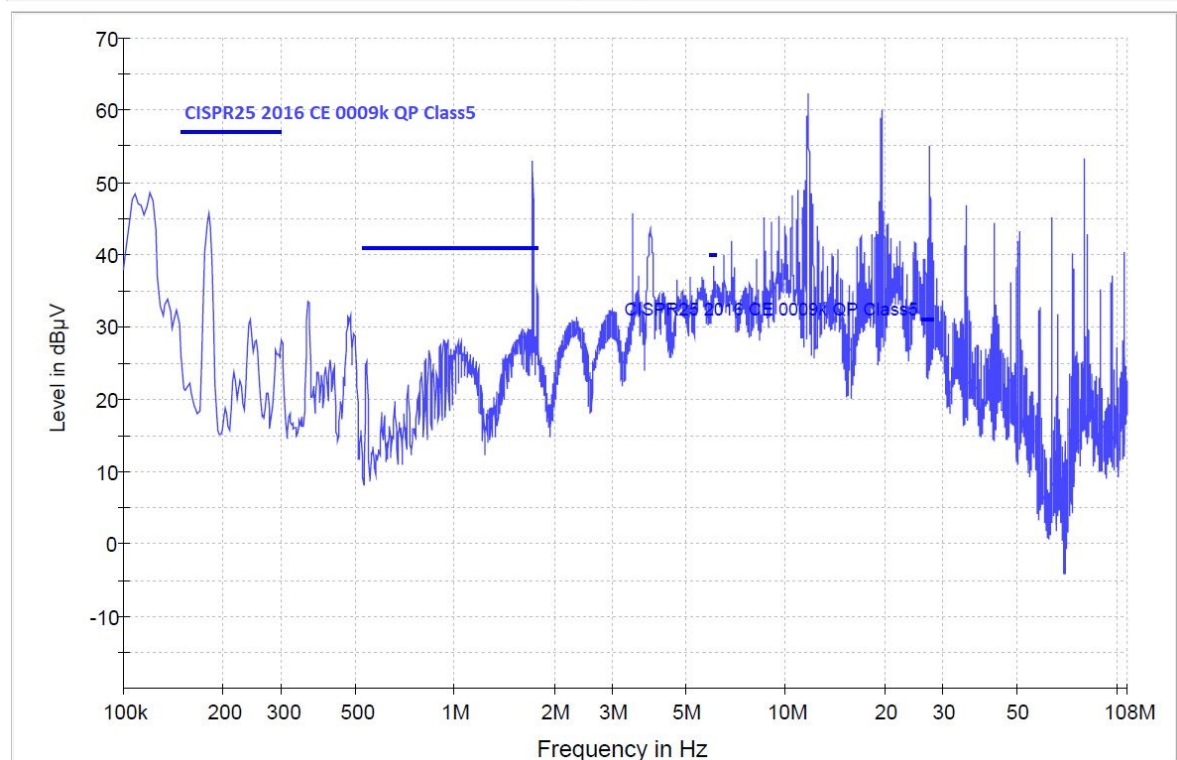
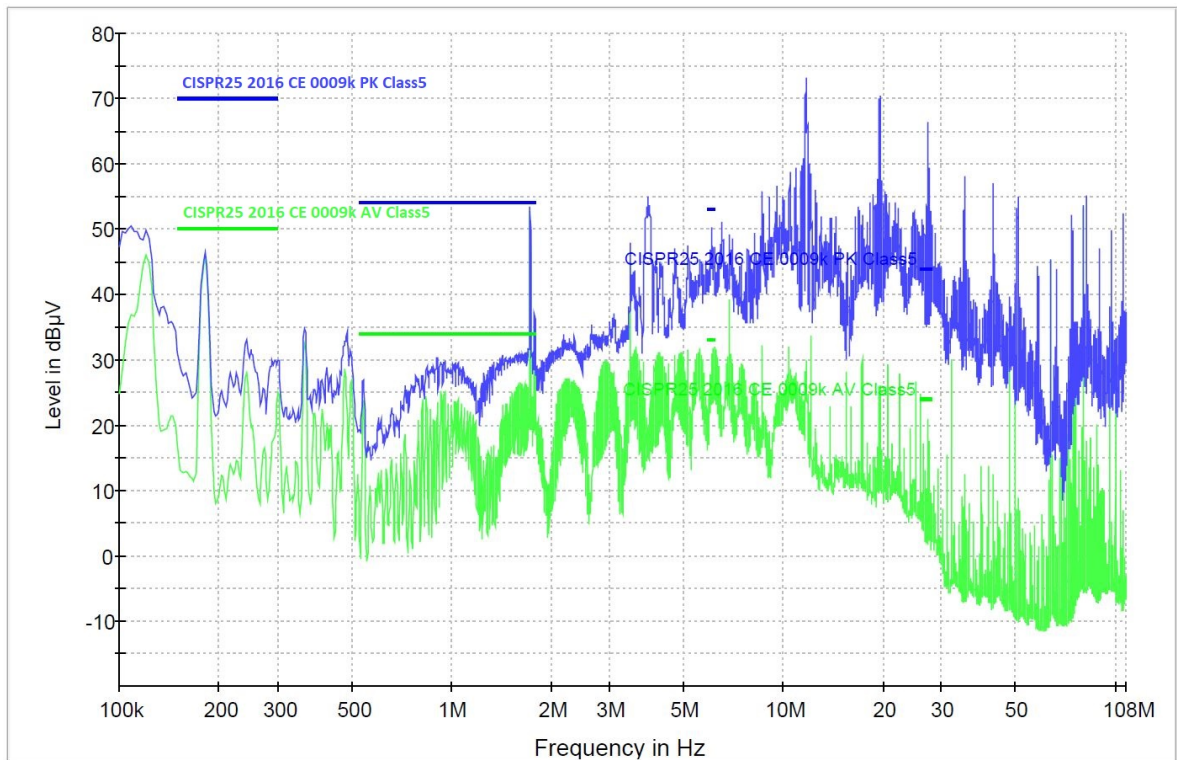
$$Level[dB\mu V] = 20 * \log(U[\mu V]) \quad (3.1)$$

Nastavení spektrálního analyzátoru pro měření jehož výsledky se nachází na Obr. 3.2:

- Frequency range: 100kHz - 108MHz
- Step size: 2,25kHz
- IF BW: 9kHz
- Detectors: PK+, AVG, QP
- Preamp: 20dB

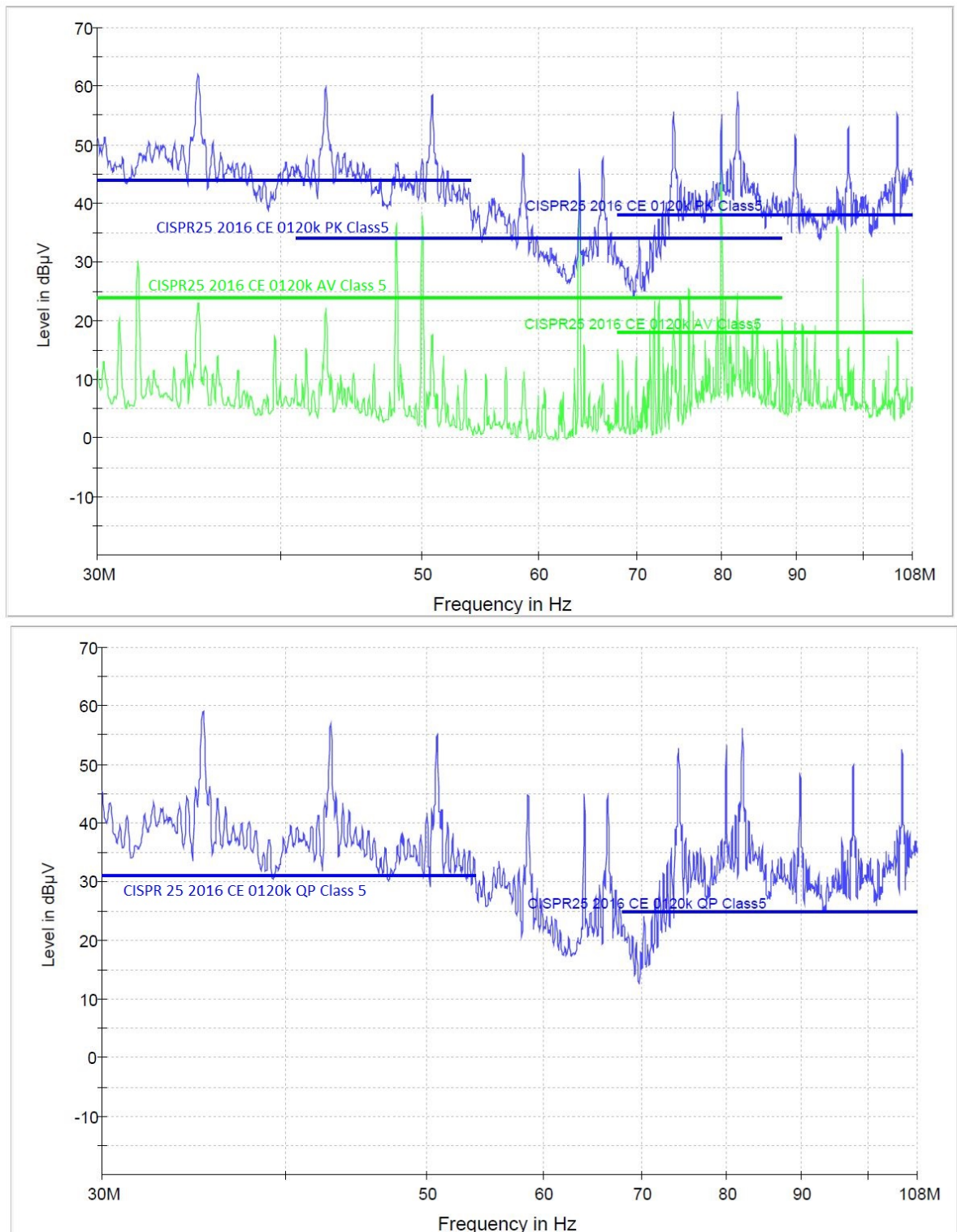
Nastavení spektrálního analyzátoru pro měření jehož výsledky se nachází na Obr. 3.3:

- Frequency range: 30MHz - 108MHz
- Step size: 30kHz
- IF BW: 120kHz
- Detectors: PK+, AVG, QP



Obr. 3.2: Výsledky měření (PK - špičková, AV - střední hodnota, QP - kvazi-špičková) pro šířku pásma mezifrekvenčního filtru 9kHz

- Preamp: 20dB



Obr. 3.3: Výsledky měření (PK - špičková, AV - střední hodnota, QP - kvazi-špičková) pro šířku pásma mezifrekvenčního filtru 120kHz

Z výsledku je vidět, že model nesplňuje limity standardu CISPR 25 pro třídu zařízení 5 ve velké části frekvenčního pásma. Je zde například vidět složku spektra na 3,9MHz, to odpovídá frekvenci signálu CLK SPI sběrnice. Jelikož se jedná o obdélníkový signál, jehož spektrum obsahuje složky na lichých násobcích základního kmitočtu, budou patrné špičky i na frekvencích $n \cdot 3,9MHz$ pro $n=3,5,7,\dots$, což odpovídá. Dále se zde mohou vyskytovat složky, které jsou parazitním jevem spínání tranzistorů.

Důvodem přenosu rušivých signálu na napájecí vedení může být velká kapacita mezi vodiči, tedy jejich blízká vzájemná poloha na desce plošného spoje. Dále může být důvodem špatné rozvedení zemnicích ploch nebo dlouhé kabely pro přívod spínaného signálu k LED diodám. Nejlepším způsobem, jak omezit rušení, by bylo jiné konstrukční řešení svítilny. Například omezení délky vodičů na desce plošného spoje, zvětšení šířky mezery mezi vodiči nebo přidání kondenzátorů k napájení spínaných prvků. Pokud by bylo zapotřebí omezit rušení bez zásahu do konstrukce modelu, bylo by vhodné přidat na napájecí vodiče blízko k lampě nějaký odrušovací prvek jako je například tlumivka, kondenzátor nebo v lepším případě odrušovací LC filtr typu dolní propust.

4 ZÁVĚR

Cílem bakalářské práce bylo vytvořit software pro platformu Raspberry Pi 3, který bude řídit model zadní svítilny automobilu, vytvořený z matice RGB LED diod. Software měl umožnit zobrazovat animace s libovolnými tvary funkcí svítilny. Úkolem bylo rovněž vytvořit uživatelské rozhraní pro platformu Windows, které umožní jak tvorbu animací, tak ovládání svítilny. Dalším úkolem práce bylo změřit elektromagnetické emise na napájecím vedení.

V první části práce bylo zapotřebí seznámit se s již hotovým demonstračním modulem RGB LED svítilny. Jsou zde popsány základní části systému: čip STLED524, komunikace po SPI rozhraní a počítač Raspberry Pi 3. Tato část rovněž obsahuje popis sběrnice CAN, která je využita pro komunikaci mezi počítačem a řídicí jednotkou modelu svítilny. Část teorie se rovněž věnuje popisu elektromagnetické kompatibility.

Praktická část obsahuje několik oddílů. Je zde upřesněno zadání projektu a jaký přístup bude použit pro realizaci. Dále jsou zde popsány funkce použité v kódu a podrobná specifikace jednotlivých bloků řídicího programu. Tato část rovněž obsahuje popis realizace komunikace CAN na Raspberry Pi 3. Jedna z kapitol se věnuje popisu vytvořeného uživatelského rozhraní pro ovládání a tvorbu animací a vysvětluje postup při jejich tvorbě. Poslední kapitola praktické části je zaměřená na vyhodnocení měření Conducted emissions.

V bakalářské práci se podařilo naplnit zadání. Stávající software umožňuje vytvořit funkce zadní svítilny libovolných tvarů s libovolnou animací. Je rovněž umožněno přepínat mezi třemi základními módy světla, tedy sport, normal a economy. Podle zadání je řízení svítilny i přenos barevných schémat z počítače realizováno přes sběrnici CAN. Tvorba animací je podle požadavků jednoduchá, ale zdlouhavá v případě tvorby animace obsahující příliš mnoho snímků. Celek tvořený modelem svítilny a řídicím počítačem RaspberryPi je otestován podle zadání na Conducted emissions.

Pro případný další vývoj projektu by šlo navrhnout několik zlepšení. Jednalo by se zejména o lepší konstrukční řešení modelu svítilny, které by zajistilo lepší funkčnost celku i lepší výsledky měření rušení na vedení. Jednalo by se tedy o použití vhodnějších čipů pro řízení LED diod, implementaci řídicího mikrokontroléru do modelu svítilny a aplikaci odrušovacích filtrů.

LITERATURA

- [1] HOFMAN, Denis. *Návrh a realizace automobilové zadní světlíny s RGB LED*. Ostrava, 2017. Diplomová práce. VŠB Ostrava. Vedoucí práce Ing. Petr Šimoník, Ph.D.
- [2] STMicroelectronics. *STLED524 Intelligent matrix LED display driver* [online]. 2014, Poslední aktualizace 2014. [cit. 2017-12-07]. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/datasheet/6d/fb/4d/93/de/80/45/b2/DM00113642.pdf/files/DM00113642.pdf/jcr:content/translations/en.DM00113642.pdf>
- [3] Serial Peripheral Interface Bus. *Wikipedia: The free encyclopedia* [online]. San Francisco: Wikimedia Foundation, 28.11.2017 [cit. 2017-12-06]. Dostupné z: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus
- [4] FRÝZA, Tomáš. *Řízení sériové komunikace: Mikroprocesorová technika a embedded systémy*. Brno, 2017. Prezentace. UREL, VUT v BRNĚ.
- [5] POLÁK, Karel. *Sběrnice CAN* [online]. 16.6.2003 [cit. 2018-02-24]. Dostupné z: <http://www.elektrorevue.cz/clanky/03021/index.html>
- [6] NEUMANN, David. *Sériová komunikace v automobilové technice: seminář v rámci předmětu BMPT*. VUT Brno, FEKT, 2017.
- [7] CAN bus. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-03-06]. Dostupné z: https://en.wikipedia.org/wiki/CAN_bus
- [8] *RASPBERRY PI 3 MODEL B: Technical specifications* [online]. Cambridge: Raspberry Pi Foundantion, 2016 [cit. 2017-12-07]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [9] DŘÍNOVSKÝ, Jiří, Tomáš FRÝZA, Václav RŮŽEK a Jiří ZACHR. *Elektromagnetická kompatibilita* [online]. VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, Technická 12, 616 00 Brno, 2017 [cit. 2018-04-22]. Dostupné z: <http://www.urel.feec.vutbr.cz/~drinovsky/?download=Skripta EMC.pdf>. Skripta. Vysoké učení technické v Brně.
- [10] SACCHI, Roberto. *Misure di compatibilita' Elettromagnetica* [online]. Italy: Electronic Measurements Group Agilent Technologies [cit. 2018-04-23]. Dostupné z: https://www.keysight.com/upload/cmc_upload/All/EMC_Compliance_Seminar.pdf

- [11] MCCAULEY, Mike. *Bcm2835: C library for Broadcom BCM 2835 as used in Raspberry Pi* [online]. 2017-02-03 [cit. 2017-12-06]. Dostupné z: <http://www.airspayce.com/mikem/bcm2835/index.html>
- [12] *<termios.h>: The Single UNIX® Specification, Version 2* [online]. The Austin Common Standards Revision Group, 1997- [cit. 2017-12-08]. Dostupné z: <http://pubs.opengroup.org/onlinepubs/007908799/xsh/termios.h.html>
- [13] ADAMCZYK, Bogdan. *Conducted Emissions Measurements: Voltage method* [online]. 2017 [cit. 2018-04-23]. Dostupné z: <https://incompliancemag.com/article/conducted-emissions-measurements-voltage-method/>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

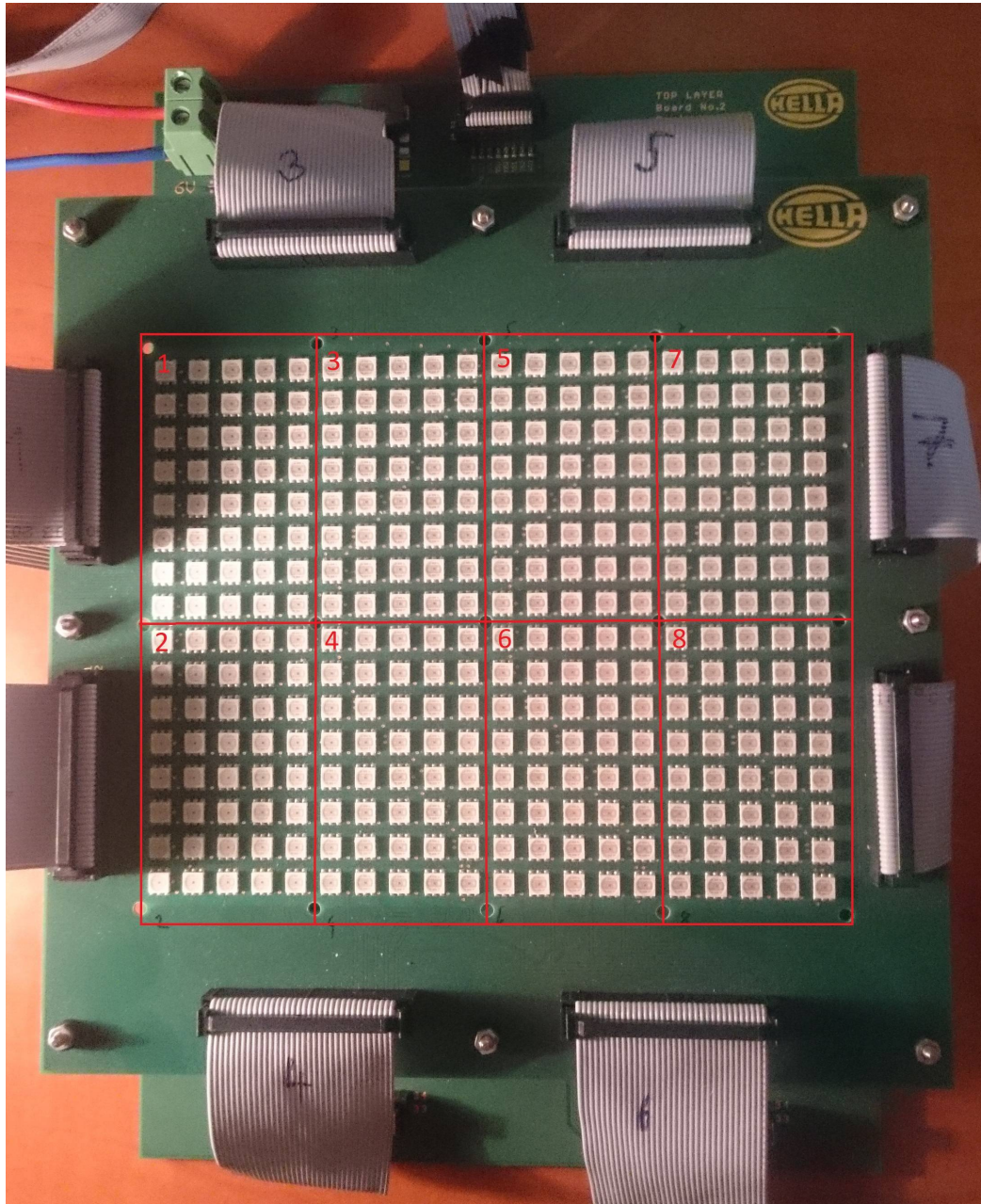
DSP	číslicové zpracování signálů - Digital Signal Processing
RGB	barevný model červená, zelená, modrá - Red, Green, Blue
LED	Světlo emitující dioda - Light Emitting Diode
HW	Hardware
RPi	Raspberry Pi
SPI	seriové periferní rozhraní - serial peripheral interface
PWM	pulzně šířková modulace - pulse width modulation
GPIO	vstupně-výstupní piny pro všeobecné účely - general purpose input/ output
CAN	Controller Area Network
EMC	elektromagnetická kompatibilita - electromagnetic compatibility
EMI	elektromagnetická interference - electromagnetic interference
EMS	elektromagnetická susceptibilita - electromagnetic susceptibility
CE	rušivé signály na vedení - conducted emissions
LISN	umělá zátěž vedení - line impedance stabilizing network
AMN	umělá síť - artificial mains network
P	špičková hodnota - peak value
QP	kvazi-špičková hodnota - quasi-peak value
AV	střední hodnota - average value

SEZNAM PŘÍLOH

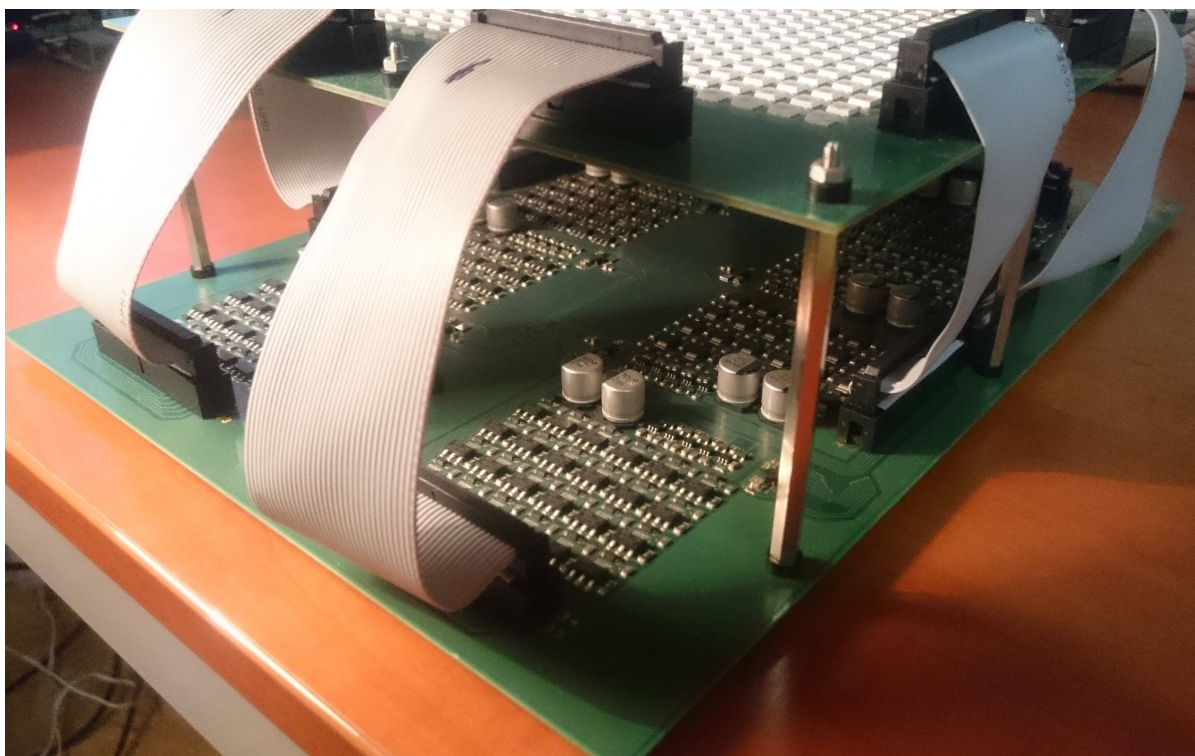
A Fotodokumentace přípravku	54
B Měření conducted emissions	56
C Obsah přiloženého CD	59

A FOTODOKUMENTACE PŘÍPRAVKU

Příloha obsahuje fotodokumentaci modelu zadní matrix LED lampy, pro kterou byl tvořen software.



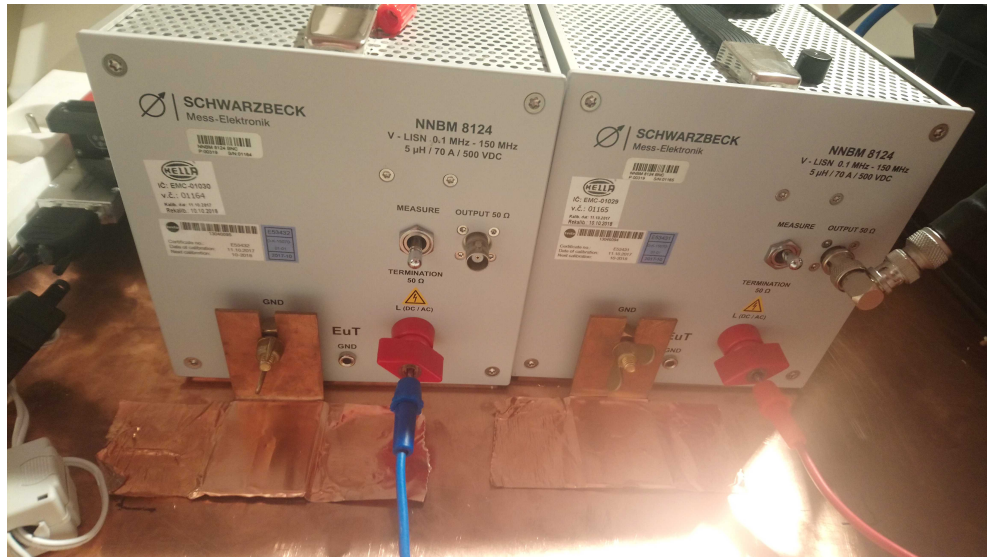
Obr. A.1: Demonstrační model svítilny, je zde naznačeno rozdělení displeje, kdy každou osminu displeje řídí jeden čip STLED524



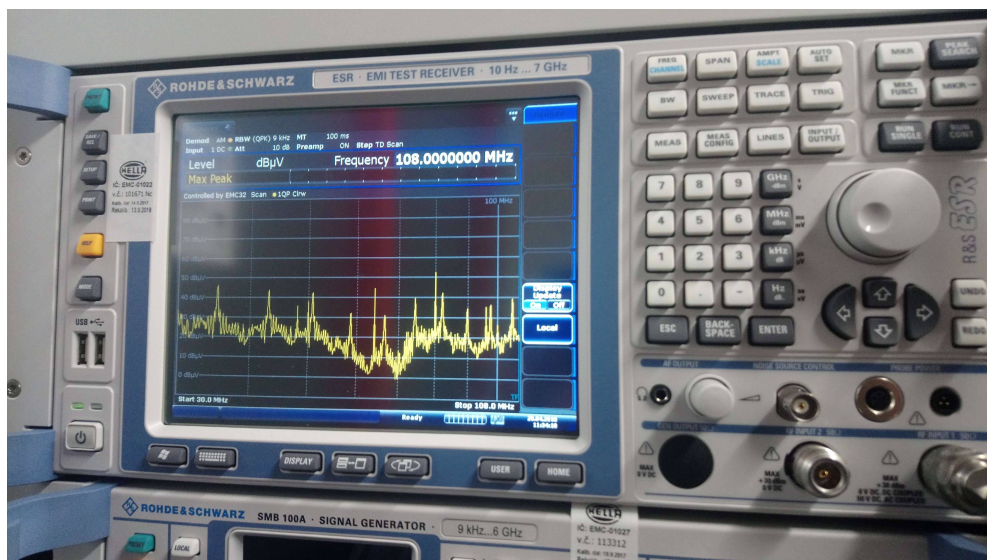
Obr. A.2: Demonstrační model RGB svítily

B MĚŘENÍ CONDUCTED EMISSIONS

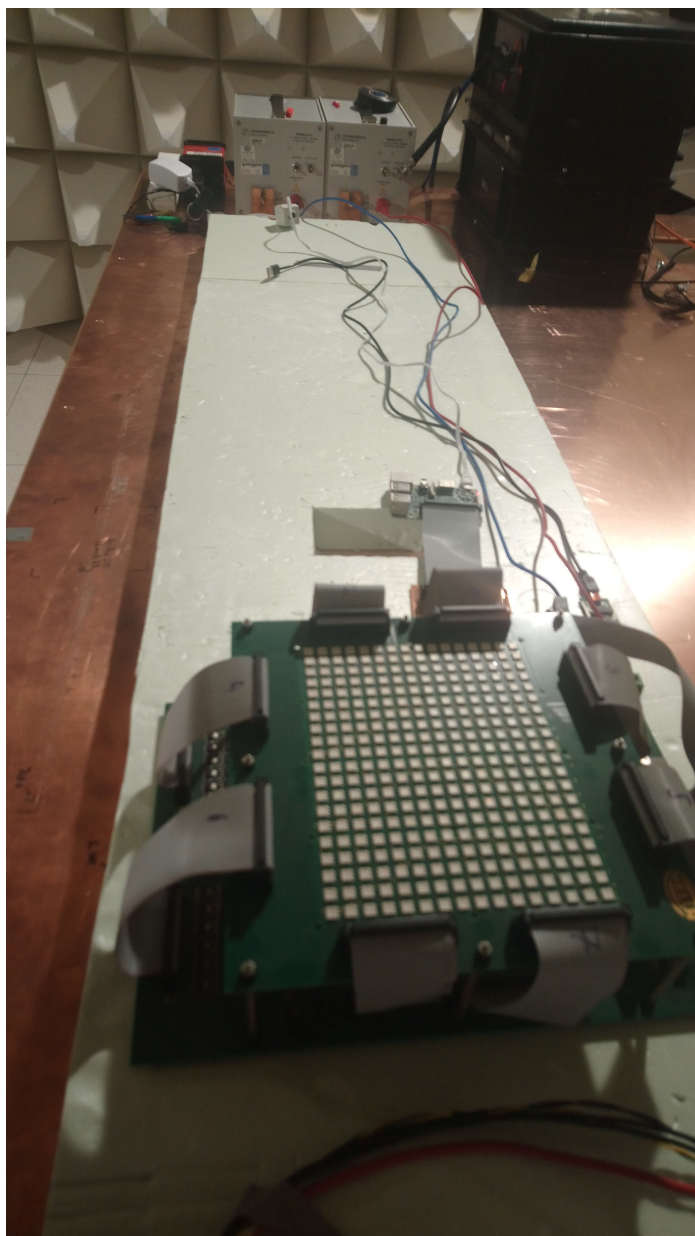
V příloze se nachází fotodokumentace pracoviště pro měření conducted emission. Na Obr. B.3 a B.4 je zapojení pracoviště podle doporučení CISPR 25. Celé měřicí pracoviště se nachází uvnitř stínící komory.



Obr. B.1: Umělá síť AMN NNBM 8124 použitá v měření, kdy spektrálním analyzáto-rem je snímána pouze umělá síť připojená na kladný napájecí vodič.



Obr. B.2: Spektrální analyzátor Rohde&Schwarz ESR7 řízený počítačem



Obr. B.3: Uspořádání měřícího pracoviště 1



Obr. B.4: Uspořádání měřícího pracoviště 2

C OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje dvě složky. Ve složce *RPI* se nachází jak kompletní řídicí software pro počítač Raspberry Pi tvořený v programu Geany, tak soubor potřebný pro přiřazení CAN modulu ke sběrnici SPI 1. Další složka *Windows* obsahuje projekt programu SharpDevelop, který představuje uživatelské rozhraní pro platformu Windows. Je zde rovněž uloženo několik světelných animací v textových souborech. Podrobný obsah CD je možné vidět na *Obr. C.1*.



Obr. C.1: Stromový výpis obsahu příloženého CD