



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**MODULÁRNÍ ZOBRAZOVACÍ SYSTÉM NA BÁZI LED
DIOD**

MODULAR DISPLAY SYSTEM WITH LED DIODES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JIŘÍ FŇUKAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Fňukal Jiří**

Obor: Informační technologie

Téma: **Modulární zobrazovací systém na bázi LED diod
Modular Display System with LED Diodes**

Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se s architekturou a programovacím modelem rodiny mikrokontrolérů Kinetis.
2. Provedte rešerši existujících průmyslových komunikačních sběrnic, přičemž pozornost zaměřte na sběrnici CAN.
3. Navrhněte koncepci zobrazovacího systému, který je tvořen moduly na bázi LED diod vzájemně propojitelnými do větších celků. Systém bude vybaven komunikačním rozhraním Bluetooth.
4. V návrhovém editoru připravte realizaci desky plošných spojů modulu s LED diodami. Několik těchto modulů fyzicky zhotovte.
5. Vytvořte program pro architekturu Arduino, která bude fungovat jako master pro celý display složený z dílčích modulů.
6. Vhodným způsobem demonstруйте funkčnost vámi navrženého řešení.
7. Zhodnoťte dosažené výsledky a pokuste se navrhnout případná rozšíření.

Literatura:

- Noergaard, T. Embedded Systems Architecture, Elsevier, Burlington, 2005, 657 s., ISBN 0-7506-7782-9.
- Axelson, J. Embedded Ethernet and Internet Complete, Lakeview Research, Madison, 2003, 497 s., ISBN 1-931448-01-9.
- Williams, T. The Circuit Designer's Companion, Elsevier, New York, 354 s., ISBN 0-7506-6370-7

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Šimek Václav, Ing.**, UPSY FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2



prof. Ing. Lukáš Sekanina, Ph.D.
vedoucí ústavu

Abstrakt

Cílem této bakalářské práce je vytvoření Modulárního zobrazovacího systému na bázi LED diod. Práce je rozdělena do tří částí. První část popisuje dostupné LED systémy, které lze sehnat na trhu, obsahuje také popis vývojových platforem. V druhé části této práce je proveden rozbor průmyslových sběrnic, s důrazem na CAN sběrnici. Poslední celek shrnuje celkový návrh a konstrukci zobrazovacího systému. Po přečtení této práce by se měl čtenář orientovat v LED systémech, průmyslových komunikačních sběrnicích a měl by být schopen tento zobrazovací systém sestavit.

Abstract

The purpose of bachelor theses is to create a modular display system with LED diodes. The work is divided into three main section. The first section deals with the description of the LED systems available on the market, also contains a description of development platforms. The second part of this work focuses on the industrial buses, with an emphasis on the CAN bus. In the last section is described the design and construction of the LED system. After reading this work, the reader should be able to navigate in LED systems, industrial communication buses and should be able to construct this LED system.

Klíčová slova

Velkoplošná obrazovka, LED panel, Arduino, Raspberri Pi, Android, CAN BUS, Bluetooth, SPI, sběrnice, modul, LED dioda

Keywords

Video wall, LED panel, Arduino, Raspberri Pi, Android, CAN BUS, Bluetooth, SPI, bus, modul, LED diode

Citace

FŇUKAL, Jiří. *Modulární zobrazovací systém na bázi LED diod*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

Modulární zobrazovací systém na bázi LED diod

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Václava Šimka. Další informace o výrobě plošných spojů mi poskytl Ing. Jan Hána. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jiří Fňukal
17. května 2017

Poděkování

Chtěl bych poděkovat vedoucímu práce, panu Ing. Václavu Šimkovi za cenné rady při tvorbě bakalářské práce. Dále mé poděkování patří panu Ing. Janu Hánovi, který mi poskytl prostor pro výrobu desek plošných spojů a také za jeho cenné rady při tvorbě těchto desek.

Obsah

1	Úvod	3
2	Velkoplošné LED zobrazovače	4
2.1	ADJ AV6X	4
2.2	CHAUVET F4	5
2.3	G-LEC Phantom 30	6
2.4	ELATION EZ4	7
2.5	PLANAR NVH08S	8
2.6	PixelFLEX FLEXStorm	9
3	Vývojové platformy	11
3.1	Arduino	11
3.1.1	Historie	11
3.1.2	Programování	11
3.1.3	Arduino Nano	11
3.2	Raspberry Pi	12
3.2.1	Historie	13
3.2.2	Programování	13
3.2.3	Raspberry Pi 3 Model B	13
4	Průmyslové sběrnice	15
4.1	CAN	15
4.1.1	Základní vlastnosti protokolu	16
4.1.2	Fyzická vrstva	17
4.1.3	Linková vrstva	18
4.1.4	Typy zpráv	19
4.1.5	AS-I (<i>AS-Interface</i>)	22
4.1.6	DeviceNet	23
4.1.7	InterBus	23
4.1.8	ProfiBus	24
5	Konečný návrh	26
5.1	Vlastnosti	26
5.2	Koncepce	27
6	Hardwarové řešení	28
6.1	LED modul	28
6.1.1	Ovládací prvek	28

6.1.2	CAN BUS modul	28
7	Programové řešení	30
7.1	LED modul	30
7.1.1	Arduino	30
7.1.2	CAN BUS rozhraní	30
7.1.3	Ovládání displeje	32
7.2	Řídící modul	33
7.2.1	Hlavní programová smyčka	33
7.2.2	Bluetooth rozhraní	33
7.2.3	CAN BUS rozhraní	34
7.3	Aplikace Android	34
7.3.1	Hlavní aktivita	34
7.3.2	Připojení	34
7.3.3	Nastavení	35
7.3.4	O aplikaci	35
8	Testování	37
8.1	Testování modulárního zobrazovacího systému	37
8.2	Možnosti rozšíření	38
9	Závěr	39
	Literatura	40
A	Schéma zapojení LED modulu	42
B	Deska plošných spojů – spoje	44
C	Deska plošných spojů – rozložení součástek	45
D	Osazený LED modul	46
E	Výsledný celek	47

Kapitola 1

Úvod

Cílem této bakalářské práce je popis konstrukce modulárního zobrazovacího systému na bázi LED diod. Na začátku se práce zaměřuje na již dostupné zobrazovací LED systémy, které lze sehnat na trhu. Jsou zde shrnuty jejich hlavní vlastnosti a popsány možnosti řešení. Další důležitou částí je rešerše vývojových platforem a průmyslových komunikačních sběrnic, kde je důraz kladen na popis CAN sběrnice. Díky těmto poznatkům je cílem sestavit vlastní zobrazovací systém, který bude tvořen z jednotlivých LED modulů, bude rozšiřitelný do větších celků a který bude možno ovládat pomocí Bluetooth rozhraní.

Práce také obsahuje návrh modulárního zobrazovacího systému, kde jsou přiblíženy jeho vlastnosti a sestavení dílčích částí. Kapitola hardwarové řešení je zaměřena na jednotlivá schémata a popis jejich funkčnosti. Navazující část softwarové řešení nastiňuje programové řešení jednotlivých modulů a popisuje komunikaci se sběrnicí CAN, obsluhu LED displeje a Bluetooth komunikaci. V předposlední kapitole je popsáno testování, které probíhalo po celou dobu konstrukce a jeho výsledky.

Hlavním důvodem k výběru této bakalářské práce je obliba práce s elektronikou, navrhování nových věcí, tvorba hardwaru a práce s LED diodami.

Kapitola 2

Velkoplošné LED zobrazovače

Velkoplošné LED obrazovky jsou v současné době nejvíce využívány v oblasti reklam, ale také v mnoha jiných odvětvích. Jsou to displeje složené z tisíců RGB LED diod, které jsou rozloženy do jednotlivých celků. Většina podporuje prezentaci obrázků nebo přehrávání videa v plné barevné škále (16,7 milionů barev). Tyto zobrazovače se vyrábí pro vnitřní i vnější použití. Níže jsou popsány jejich vlastnosti, které nám nabízejí.

2.1 ADJ AV6X

Tento zobrazovací systém, který můžeme vidět na obrázku 2.1 je vhodný jako velkoplošný LED zobrazovač. Jednotlivé body jsou tvořeny RGB diodami *SMD 2121*. Tyto LED diody přinášejí vynikající barevnou konzistenci a je použito tmavé maskování pro vyšší kontrast. Tento panel je připojen konektory *PowerCon I/O* a *EtherCon I/O*, spolu s *Navistar* přijímací kartou. AV6X je složen ze čtyř menších modulů, které tvoří jeden celek. Díky těmto vlastnostem poskytuje snadnou instalaci a obsluhu. [1]

- **LED konfigurace:** RGB SMD 2121
- **Rozteč pixelů:** 6 mm
- **Počet pixelů:** 9216 (96 x 96)
- **Jas:** 1200 Nits (0 – 100 %)
- **Obnovovací frekvence:** 1200 Hz
- **Barva displeje:** 14 bitů
- **Pozorovací úhel:** 110°
- **Pozorovací vzdálenost (minimální):** 6 m
- **Příkon (maximální):** 300 W
- **Hmotnost:** 23 kg
- **Rozměry:** 576 x 576 mm
- **Konektory:** PowerCon I/O, EtherCon I/O



Obrázek 2.1: Přední a zadní pohled na ADJ AV6 [1]

2.2 CHAUVET F4

F4 je 1 m x 0,5 m velký LED panel, který je ideální pro různé druhy vnitřních instalací. Komunikace probíhá na protokolu *Novastar*, který dovoluje řídit až 120 jednotlivých panelů a celkový počet ovládaných LED je 2621440. Ke snadné instalaci slouží magnetické LED moduly, které jsou pomocí magnetu přichyceny k rámu. K zobrazování jsou použity *RGB SMD 2528*, které zajišťují velký pozorovací úhel s vysokou svítivostí. Tento displej se nachází na obrázku 2.2. [4]

- **LED konfigurace:** RGB SMD 2828
- **Rozteč pixelů:** 4,8 mm
- **Počet pixelů:** 21632 (104 x 208)
- **Jas:** 1500 Nits
- **Obnovovací frekvence:** 3840 Hz
- **Pozorovací úhel:** 140°
- **Pozorovací vzdálenost (minimální):** 4 m
- **Příkon (maximální):** 248 W
- **Hmotnost:** 12,88 kg
- **Rozměry:** 1000 x 500 mm
- **Konektory:** PowerCon I/O, EtherCon I/O



Obrázek 2.2: Přední a zadní pohled na CHAUVET F4 [4]

2.3 G-LEC Phantom 30

Phantom 30 je systém navržený pro touring, je tvořen hliníkovými rámy, které se upevňují systémem *QuickLock*. To zajišťuje rychlou a pohodlnou instalaci. Díky transparentnosti umožňuje displej průchod světla, zvuku a vzduchu přes jednotlivé moduly. Více modulů je možné skládat do větších celků a díky lehké váze lze zavěsit tento displej i na klasické konstrukce bez speciálního vyztužení. Jednotlivé panely jsou napájeny externími zdroji a řízeny *Driver Boxem*, který komunikuje s PC. Komunikace mezi LED panely a *Driver Boxem* probíhá pomocí optického kabelu, kterým jsou propojeny i jednotlivé moduly mezi sebou. LED zobrazovač je použitelný pro vnější i vnitřní použití a zobrazen na obrázku 2.3. [9]

- **LED konfigurace:** RGB SMD
- **Rozteč pixelů:** 30,0 mm
- **Počet pixelů:** 1024
- **Jas:** 6220 Nits
- **Transparentnost:** 60
- **Obnovovací frekvence:** 300 Hz
- **Pozorovací úhel:** 110°
- **Pozorovací vzdálenost (minimální):** 4 m
- **Příkon (maximální):** 420 W

- **Hmotnost:** 8 kg
- **Rozměry:** 960 x 960 mm
- **Konektory:** optický kabel



Obrázek 2.3: Přední pohled na G-LEC Phantom 30 [9]

2.4 ELATION EZ4

Jedná se o LED video panel s vysokým rozlišením a kontrastem určený k promítání obrazu. Díky jeho vysoké svítivosti je vhodný i pro práci za denního světla. Je opatřen RGB diodami *SMD 2121 Black Face* a je určen pro použití v interiéru či pro stálou instalaci. Tento panel je možné vidět na obrázku 2.4. [6]

- **LED konfigurace:** SMD 2121 Black Face
- **Rozteč pixelů:** 4,81 mm
- **Počet pixelů:** 10816 (104 x 104)
- **Jas:** 1200 Nits
- **Kontrast:** 1000:1
- **Barva displeje:** 16 bitů
- **Obnovovací frekvence:** 1200 Hz

- **Pozorovací úhel:** 110°
- **Příkon (maximální):** 230 W
- **Hmotnost:** 10 kg
- **Rozměry:** 500 x 500 mm



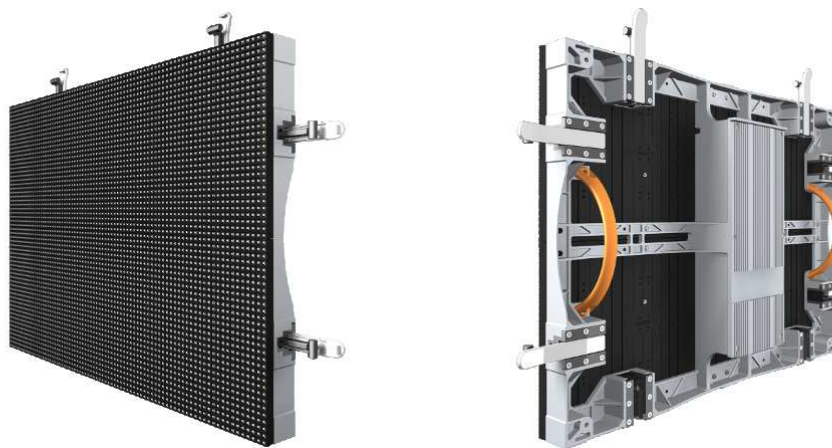
Obrázek 2.4: Přední a zadní pohled na ELATION EZ4 [6]

2.5 PLANAR NVH08S

Jedná se o komplexní systém pro tvorbu LED video displeje zobrazený na obrázku 2.5. Lze ho použít pro vnitřní i venkovní použití a odolá i nepříznivým povětrnostním podmínkám. Má široký pozorovací úhel. Tento systém je vhodný pro širokou škálu aplikací a zajišťuje dlouhou životnost. [12]

- **LED konfigurace:** RGB SMD
- **Rozteč pixelů:** 8,0 mm
- **Počet pixelů:** 4800 (80 x 60)
- **Jas:** 5500 Nits
- **Kontrast:** 5000:1
- **Obnovovací frekvence:** 5100 Hz
- **Pozorovací úhel:** 160°
- **Příkon (maximální):** 248 W

- **Hmotnost:** 11 kg
- **Rozměry:** 640 x 480 mm
- **Konektory:** RS232, Ethernet



Obrázek 2.5: Přední a zadní pohled na PLANAR NVH08S [12]

2.6 PixelFLEX FLEXStorm

Tento systém je navržen pro venkovní použití, tak aby odolal nepříznivým povětrnostním podmínkám. LED panel je vhodný například pro reklamu či zábavu a jeho vysoká svítivost zajišťuje dobrou čitelnost ve dne i v noci. Díky mnoha senzorům, které sledují venkovní prostředí, je zajištěna nejlepší čitelnost během celého dne. Pracuje s komunikačním protokolem *NovaStar* a lze ho vidět na obrázku 2.6. [11]

- **LED konfigurace:** RGB SMD 3030
- **Rozteč pixelů:** 6,67 mm
- **Počet pixelů:** 21632 (144 x 144)
- **Jas:** 5000 Nits
- **Obnovovací frekvence:** 1920 Hz
- **Pozorovací úhel:** 140°
- **Pozorovací vzdálenost (minimální):** 1,79 m
- **Příkon (maximální):** 900 W
- **Hmotnost:** 30 kg
- **Rozměry:** 960 x 960 mm



Obrázek 2.6: Přední a zadní pohled na FLEXStorm [11]

Kapitola 3

Vývojové platformy

3.1 Arduino

Arduino je založeno na procesorech od firmy Atmel, kde každý je obklopen spoustou dalších elektronických komponent. Jako je například převodník umožňující komunikaci mezi PC a procesorem pomocí rozhraní USB. Tento převodník se používá například k nahrání programu na vývojovou desku. Existují i desky bez převodníku, důvodem je buď uspořené místo, nebo již má procesor tento převodník v sobě zabudovaný. [16]

3.1.1 Historie

Vývoj prvního *Arduina* započal v roce 2005 v *Interaction Design Institute* v Itálii. Jeho cílem bylo vytvořit jednoduchý a levný vývojový kit pro studenty. Od té doby se *Arduino* uchytilo a bylo roku 2010 zveřejněno celému světu. Programová část *Arduina* byla založena na *Processing*, což je programovací jazyk s vlastním editorem. Do dnešní doby vzniklo spoustu různých druhů desek. Protože se jedná o opensource projekt, vzniklo i mnoho dalších neoficiálních typů (klonů), mezi které patří například *Seeeduino*, *Freaduino* nebo *ArduPilot*. [16]

3.1.2 Programování

Arduino lze programovat v jazyce C nebo C++. Nejjednodušší je ale používat knihovnu *Wiring*, která je hojně využívána. Díky její komplexnosti se o ní dá mluvit jako o samostatném programovacím jazyce. Základem každého programu jsou dva bloky programu které můžeme vidět na obrázku 3.1. Kde v horní části vidíme funkci `void setup()`, do které se píše kód, který je proveden pouze jednou na začátku programu. Zde se může nacházet například inicializace různých prvků, či rozhraní se kterými se pracuje. Druhým blokem je funkce `void loop()`, zde se píše kód, který se provádí neustále dokola až do odpojení napájení. [16]

3.1.3 Arduino Nano

Arduino Nano je osazeno procesorem *ATmega328* s taktem 16 MHz. Také obsahuje Mini-B USB konektor a převodník (*CH340*). Používá se v konstrukcích, kde je zapotřebí úspora místa a není třeba tak velkého výpočetního výkonu. Popis desky a pinů je zobrazen na obrázku 3.2. [16]

```

1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8
9 }

```

Obrázek 3.1: Základní bloky programu pro Arduino [16]

Technické údaje:

- Mikroprocesor Atmel ATmega328 (16 MHz)
- FLASH 32 kB (z toho 2 kB zavaděč)
- SRAM 2 kB
- EEPROM 1 kB
- Napájení 7-12 V (pin Vin)
- Provozní napětí 5 V
- 14 digitálních I/O pinů (z toho 6 podporuje PWM)
- 8 vstupních analogových pinů
- Proudové zatížení pinu 40 mA

Sériový kanál: Piny 0 (RX) a 1 (TX), používají se pro příjem nebo odesílání TTL dat.

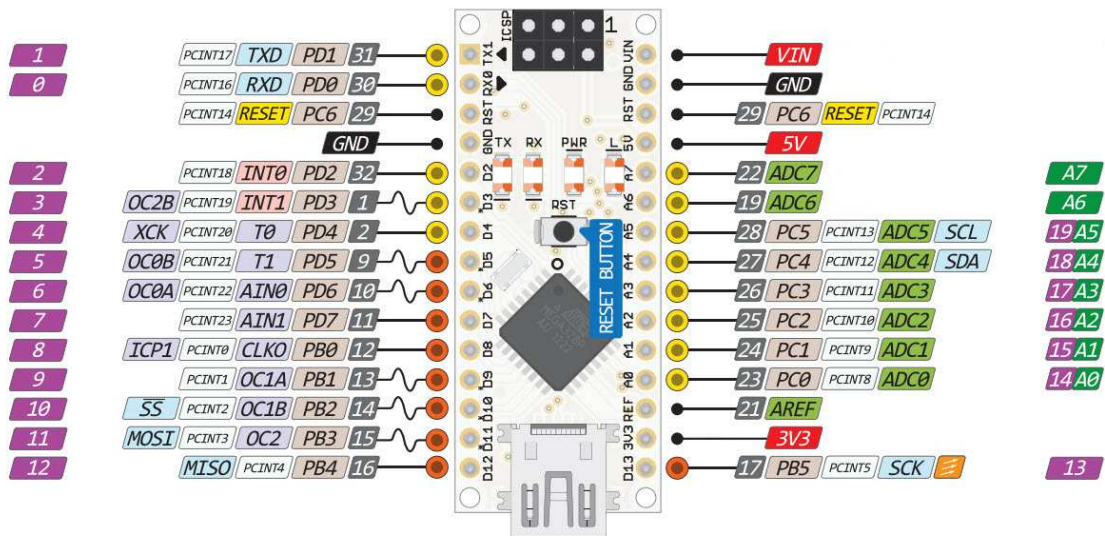
PWM: Piny 3, 5, 6, 9, 10 a 11 poskytují 8bitový PWM výstup, lze je používat funkcí `analogWrite()`.

SPI: Piny 10 (SS), 11 (MOSI), 12 (MISO) a 13 (SCK), slouží ke komunikaci s periferiemi podporujícími SPI rozhraní.

I2C: Piny A3 (SDA) a A4 (SCL), podporují I2C komunikaci.

3.2 Raspberry Pi

Raspberry Pi je jednočipový počítač založený na architektuře ARM, který je srovnatelný se slabším PC. Obsahuje několik portů USB pro připojení nejrůznějších komponent (klávesnice, myš, ...), HDMI výstup pro připojení monitoru, ethernetový port a nespočet dalších rozhraní. Na tomto počítači lze provozovat různě distribuce Linuxu (např. *Raspbian*, *NetBSD*, ...). Díky těmto vlastnostem je *Raspberry Pi* využitelný, jak k tvorbě aplikací, tak i například jako multimediální přehrávač. [17]



Obrázek 3.2: Popis jednotlivých pinů desky Arduino Nano [2]

3.2.1 Historie

V roce 2012 byla vyvinuta britskou organizací *Raspberry Pi Foundation* první verze tohoto počítače. Kladla si za cíl podpořit výuku informatiky ve školách a ukázat jak tyto malé počítače umí řídit nejrůznější zařízení, třeba automatickou pračku. V dnešní době je již třetí verze této platformy s názvem *Raspberry Pi 3*. [17]

3.2.2 Programování

Jelikož na tomto počítači běží distribuce Linuxu, je jeho programovací škála velmi pestrá. Může se zde použít téměř libovolný programovací jazyk, například *Pascal*, *Python*, *C* a v mnoho dalších. Je zde také prostředí *Scratch*, které primárně slouží k výuce programování pomocí srozumitelného a čitelného kódu. [17]

3.2.3 Raspberry Pi 3 Model B

Třetí generace Raspberry Pi je osazena 64-bitovým 4-jádrovým procesorem ARM s frekvencí 1,2 GHz. S pamětí RAM 1 GB, vestavěnou WiFi a Bluetooth viz obrázek 3.3. [17]

Technické údaje:

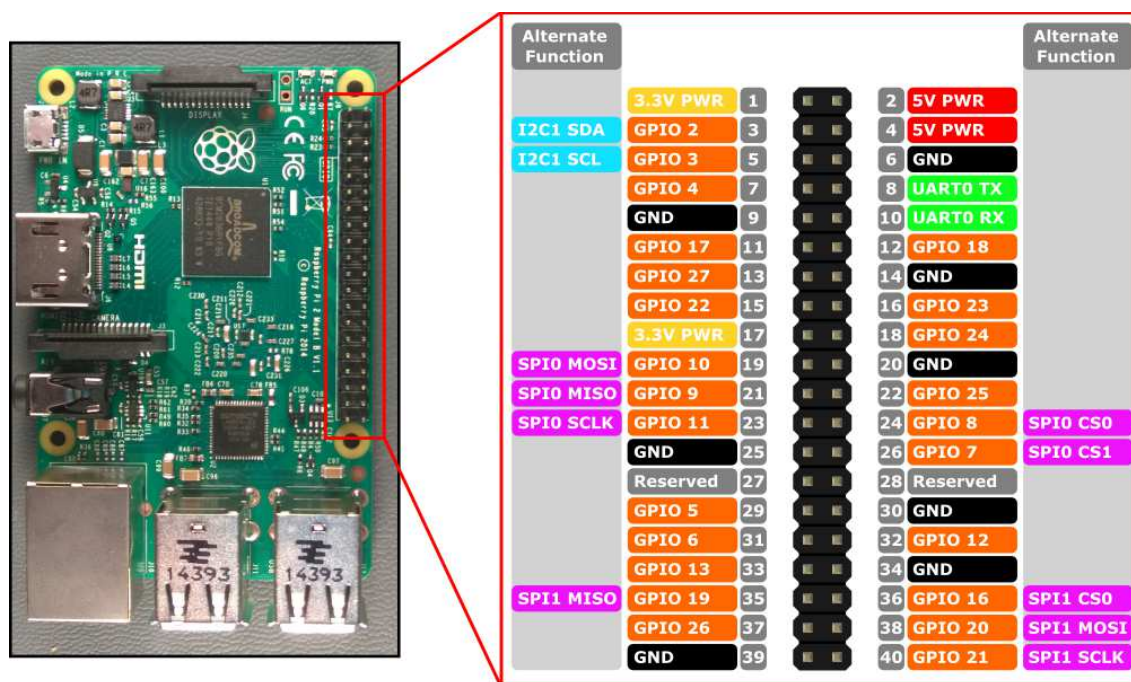
- CPU: ARM Cortex-A53 (1,2 GHz, 64-bitový, čtyřjádrový)
- GPU: Broadcom VideoCore IV
- RAM: 1GB (sdílená s GPU)
- Interní paměť MicroSDHC
- WiFi 802.11 b/g/n
- Bluetooth 4.1 BLE
- 40 x GPIO

I2C: Piny 3 (SDA) a 5 (SCL) podporují I2C komunikaci.

UART: Piny 8 (TX) a 10 (Rx) lze využít pro sériovou komunikaci.

SPI0: Piny 19 (MOSI), 21 (MISO), 23 (SCLK), 24 (CS0) a 26 (CS1) umožňují komunikaci pomocí rozhraní SPI.

SPI1: Piny 35 (MISO), 36 (CS0), 38 (MOSI) a 40 (SCLK) umožňují komunikaci pomocí rozhraní SPI.



Obrázek 3.3: Popis jednotlivých pinů desky Raspberry Pi 3 [10]

Kapitola 4

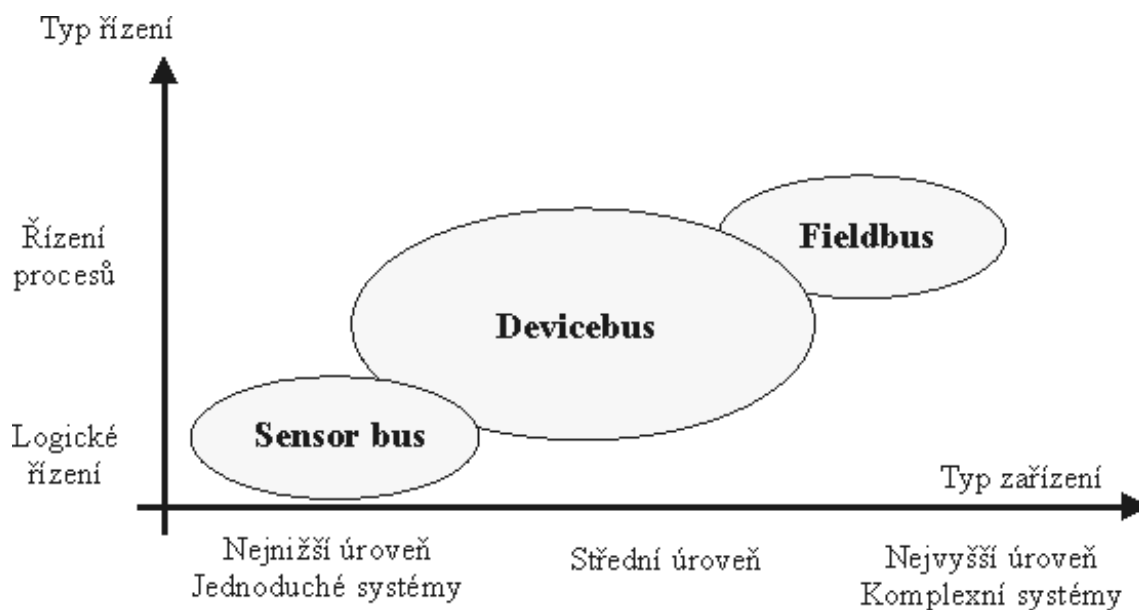
Průmyslové sběrnice

Takto kapitola je věnována průmyslovým sběrnicím, které jsou v této době velmi důležité v mnoha průmyslových odvětvích. Slouží k přenosu dat, například k měření fyzikálních veličin, či k ovládání různých elektronických systémů. Můžeme se s nimi setkat v telekomunikačním, strojírenském, či automobilovém průmyslu a jejich využití je velmi pestré. Tyto sběrnice se podobají obyčejným lokálním počítačovým sítím. Rozdílem je fakt, že jsou na ně kladeny nároky na zajištění řízení v reálném čase, bezpečnost a odolnost přenosu dat při rušení, což přináší nutnou modifikaci. Na obrázku 4.1 můžeme vidět jejich rozdělení. [7]

- **Vrstva senzorů a akčních členů** (*Sensor Bus*) – používá se ke komunikaci s akčními členy a senzory v reálném čase. Tyto sítě pracují na nejnižší úrovni a většinou definují pouze 1. a 2. vrstvu ISO/OSI modelu. Díky této vlastnosti jsou jednoduchou a efektivní technologií, kde jsou data přenášena s napájením o společném komunikačním médiu. Probíhá zde komunikace pomocí binárních signálů. Příkladem takovéto sítě jsou: *AS-Interface*, *Interbus* a další.
- **Vrstva sběrnicových zařízení** (*Devicebus*) – používá se pro komunikaci na střední úrovni. Slouží k propojení ovládacích a I/O zařízení v reálném čase. Tyto sběrnice definují 1., 2. a 7. vrstvu ISO/OSI modelu. Jsou u nich použity delší rámce a umožňují konfiguraci senzorů a akčních členů. Do této skupiny patří například průmyslové sběrnice *CAN*, *Interbus*, *DeviceNet* a spousta dalších.
- **Vrstva průmyslových komunikačních systémů** (*Fieldbus*) – je použita ke komunikaci na nejvyšší úrovni v rámci průmyslových sítí. V mnoha případech jde o sítě multimaster s redundancí přenosové linky, které podporují událostmi řízené služby, objektově orientované přenosy proměnných a dat. Příkladem mohou být sítě *Profibus*, či *P-Net* a další.

4.1 CAN

Controller Area Network (CAN) je sériový komunikační protokol, který byl vyvinut firmou Bosch pro automobilový průmysl v roce 1986. Díky svým vlastnostem se rychle rozšířil do různých průmyslových odvětví. Důvodem je nízká výrobní cena, vysoká přenosová rychlost, spolehlivost, dostupnost potřebných součástek a snadné nasazení. Nyní je CAN nejpoužívanější komunikační sběrnicí ve vestavěných zařízeních a můžeme se s ním setkat mimo automobilového průmyslu například v lékařské technice, průmyslové automatizaci



Obrázek 4.1: Rozdělení sběrnic [15]

a v mnoha dalších odvětvích. V roce 1993 byl standardizován pomocí normy ISO 11898 a nesl označení CAN 2.0A. Tato norma popisuje jak fyzickou tak linkovou vrstvu tohoto protokolu podle standardního modelu ISO/OSI. Zanedlouho byla vytvořena specifikace pro CAN 2.0B. Ve které lze použít dva druhy identifikátorů, standardní 11 bitový adresový identifikátor (*standard format*) nebo 29 bitový (*extended format*). V aplikační vrstvě lze najít několik vzájemně nekompatibilních standardů (*CAL*, *CAN Kingdom*, *CANopen*, *DeviceNet*). Z důvodu kompatibility byly zavedeny následující vrstvy:

- **Vrstva objektů** – stará se o příjem a odesílání zpráv, a také rozhodování, které zprávy z transportní vrstvy budou přijaty.
- **Transportní vrstva** – zde se provádí řízení rámců, kontrola a signalizace chyb, také rozhodování zda je sběrnice volná.
- **Fyzická vrstva** – stará se o přenos jednotlivých bitů po síti a je možné si zvolit její parametry tak, aby odpovídaly potřebám prostředí, ve kterém bude nasazena.

Pokud spojíme vrstvu objektů a transportní vrstvu, získáme stejné funkce a vlastnosti, které jsou popsány v ISO/OSI modelu jako linková vrstva. [13]

4.1.1 Základní vlastnosti protokolu

CAN je sériový komunikační protokol, který umožňuje řízení systémů v reálném čase a jeho propustnost se pohybuje do 1 Mbit/s. Je zde také velký důraz na zabezpečení přenosu proti různým chybám. Tento protokol je typu *multimaster* což znamená, že jako master může být kterýkoliv připojený prvek, který následně dokáže řídit práci ostatních prvků. Z tohoto důvodu není třeba řídit celou síť centrálně z jednoho uzlu a odpadá tam potřeba jednoho výkonného centrálního prvku. Díky této vlastnosti je celá síť jednodušší a spolehlivější, při výpadku jednoho či více prvků může dále fungovat zbytek sítě. Komunikace na sběrnici

probíhá pomocí prioritního systému zpráv, kde se při kolizi dvou zpráv zpracuje vždy ta s vyšší prioritou. Dva uzly mezi sebou komunikují pomocí zpráv:

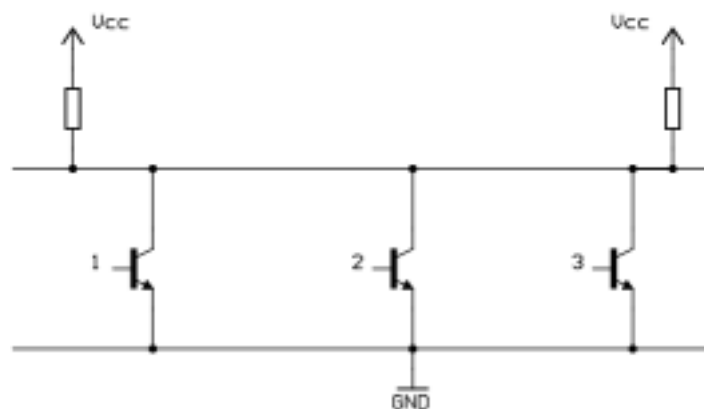
- Datová zpráva
- Žádost o data
- Síťová zpráva (stará se o signalizaci chyb)
 - Chybová zpráva
 - Zpráva o přetížení

Tyto zprávy jsou vysílány po síti a přijímány všemi připojenými prvky, které následně pomocí identifikátoru rozhodují, zde je tato zpráva určena pro tento jistý prvek sítě (zabezpečení že bude zpráva přijata pouze určitým prvkem). Pomocí identifikátoru lze také zjistit význam a prioritu přenášené zprávy. [13]

4.1.2 Fyzická vrstva

Protokol CAN je založen na ISO/OSI modelu, ale přesto definuje svoje vlastní fyzické rozhraní, které je velmi odolné vůči rušení. Je zapotřebí, aby byl tvořený spoj symetrický a fyzické médium dokázalo realizovat logický součin. Dalším standardem protokolu CAN je definice dvou hodnot bitů – *recessive* a *dominant*. Tyto bity se dají považovat za jakési logické úrovně, které se určují až při samotné realizaci fyzické vrstvy. Příkladem může být optické vlákno, které buď svítí (hodnota *dominant*), nebo nesvítí (hodnota *recessive*).

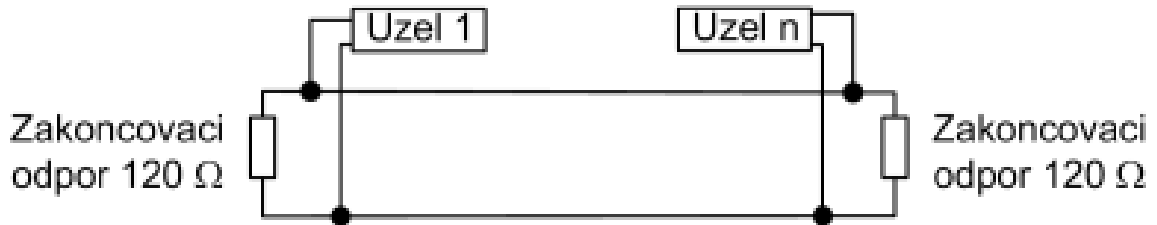
Stav na sběrnici může být tedy *dominant* nebo *recessive*. Pokud všechny uzly vysílají bit *recessive*, objeví se i tato úroveň na sběrnici. V opačném případě, pokud alespoň jeden uzel začne vysílat bit *dominant*, tato úroveň se následně objeví na sběrnici. Jako příklad by se zde dala použít sběrnice buzená hradly s otevřeným kolektorem, kterou můžeme vidět na obrázku 4.2. Pokud není sepnutý žádný tranzistor je na sběrnici logická jednička (*recessive*), v případě sepnutí jednoho či více tranzistorů se na sběrnici objeví logická nula (*dominant*). [13]



Obrázek 4.2: Realizace sběrnice pomocí tranzistorů [13]

Fyzická vrstva je většinou realizována diferenciální sběrnici definovanou normou ISO 11898. Tato norma určuje všechny důležité vlastnosti pro realizaci sběrnice (časování, synchronizaci, elektrické vlastnosti). Dále je sběrnice tvořena dvěma vodiči CAN L a CAN

H, rozdílem těchto dvou napětí zjistíme výslednou úroveň, která se ocitne na sběrnici. Dle normy pro úroveň *dominant* je rozdílové napětí $V_{diff} = 2V$ a pro úroveň *recessive* $V_{diff} = 0V$. Sběrnice je také opatřena zakončovacími odpory $120\ \Omega$, které slouží pro eliminaci odrazů. Uspořádání sítě CAN je zobrazeno na obrázku 4.3. [13]



Obrázek 4.3: Uspořádání sběrnice CAN [13]

Sběrnice může obsluhovat teoreticky libovolný počet uzlů. V reálném nasazení se uvádí okolo 64 zařízení připojených na stejnou sběrnici. Teoretická přenosová rychlost 1Mbit/s je dosažitelná pouze do vzdálenosti 40m a při delších vzdálenostech se tato rychlost velmi rychle snižuje. Díky této vlastnosti byla sběrnice CAN určena zvláště pro automobilový průmysl. [13]

4.1.3 Linková vrstva

Tato vrstva se skládá z dalších 2 částí:

- **MAC** (Medium Access Control) – provádí vkládání doplňkových bitů, zajišťuje komunikaci, řídí přístup uzlů k médiu podle priorit, kóduje zprávy, detekuje chyby a podává hlášení o chybách a přijatých zprávách.
- **LLC** (Logical Link Control) – stará se o filtrování (*Acceptance Filtering*) přijatých zpráv a hlášení o přetížení (*Overload Notification*).

Řízení přístupu ke sběrnici

Sběrnice CAN je typu *multimaster*, to znamená, že vysílání může zahájit jakýkoliv uzel ale pouze pokud je sběrnice volná (nikdo jiný právě nevysílá). Vždy může vysílat pouze jeden uzel a až poté další. Jedinou výjimkou jsou chybové zprávy, které mohou být odvysílány hned poté, co nastane chyba.

Pokud nastane stav, kdy začne vysílat více uzlů najednou, získává sběrnici uzel s vyšší prioritou (nižším identifikátorem ve zprávě). Všechny vysílače porovnávají hodnotu vysílaného bytu s hodnotou na sběrnici, a pokud se liší, okamžitě přeruší svoje vysílání. Tím je zajištěno, že nedojde k poškození vysílené zprávy a je správně doručena zpráva s vyšší prioritou. Uzel, který nezískal přístup ke sběrnici může vysílat, až je sběrnice opět volná. [13]

Zabezpečení přenosu dat

Protokol CAN má velmi silné zabezpečení přenášených zpráv a používá k tomu pět různých mechanismů. [13]

- **Monitoring** – Každý prvek porovnává vysílanou hodnotu s úrovní na sběrnici, v případě shody vysílá pořád dál. V opačném případě, pokud je na sběrnici detekována jiná úroveň a současně je vysílán *Arbitration Field* (řízení přístupu na sběrnici), přeruší se vysílání a vysílá prvek s nejvyšší prioritou. Pokud je zjištěna jiná úroveň mimo *Arbitration Field* nebo v potvrzení zprávy (ACK Slot), je generována chyba bitu.
- **CRC kód** (*Cycle Redundancy Check*) – Je dlouhý 15 bitů a je umístěn za vysílanými daty. Díky tomu může generovat kód ze všech odeslaných bitů zprávy. Využívá k tomu polynom: $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$. V případě chyby je generována chyba CRC.
- **Vkládání bitů** (*bit stuffing*) – Pokud se na sběrnici objeví pět po sobě jdoucích bitů stejné úrovně, je do zprávy vložen byt opačné úrovně. Tento mechanismus souží k detekci chyb ale také k synchronizaci připojených uzlů. V případě chyby je generována chyba vkládání bitů.
- **Kontrola zprávy** (*message frame check*) – Zpráva je kontrolována podle specifikace, která určuje povolené hodnoty na jednotlivých pozicích v rámci (zprávě). Pokud se zde objeví nepovolená hodnota, je generována chyba rámce.
- **Potvrzení přijaté zprávy** (*acknowledge*) – Každou přijatou zprávu musí každý připojený uzel (i ten který má zapnuté filtrování) potvrdit. Stává se tak změnou hodnoty v bitu ACK z hodnoty *recessive* (nastavuje vysílač) na *dominant* (nastavují příjemci).

4.1.4 Typy zpráv

Protokol CAN specifikuje tyto čtyři druhy zpráv, pomocí kterých se komunikuje:

- Datová zpráva
- Žádost o data
- Zpráva o chybě
- Zpráva o přetížení

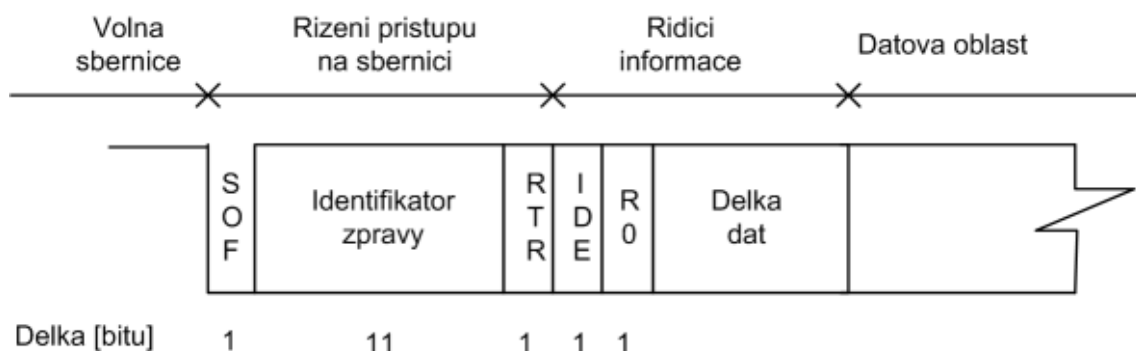
První dvě zprávy slouží pro přenos dat a jako základní komunikace na sběrnici, lze odeslat až 8 Byte dlouhou zprávu. Opačným případem jsou krátké zprávy například pro zapnutí nebo vypnutí, které nepotřebují žádná data a mohou být obsaženy pouze v identifikátoru zprávy.

Další dvě zprávy se starají o řízení sběrnice a to buď k signalizaci chyby a její odstranění, nebo k signalizaci o přetížení a vytvoření pauzy v přenosu. [13]

Datová zpráva

Na sběrnici CAN ve verzi 2.0B lze objevit dva různé typy datových zpráv. Buď standardní formát, který má identifikátor v délce 11 bitů, nebo rozšířený formát s identifikátorem v délce 29 bitů. V případě sběrnice verze 2.0A je podporován pouze standardní formát.

Zprávu lze začít vysílat, pouze pokud je sběrnice volná (*Bus Free*). Jakmile tento stav nastane a je volná, začne vysílat ten uzel, který má připravenou zprávu. Dále záleží na již popsaném chování, zda získá tento uzel přístup k médiu. Struktura zprávy verze 2.0A se nachází na obrázku 4.4. [13]

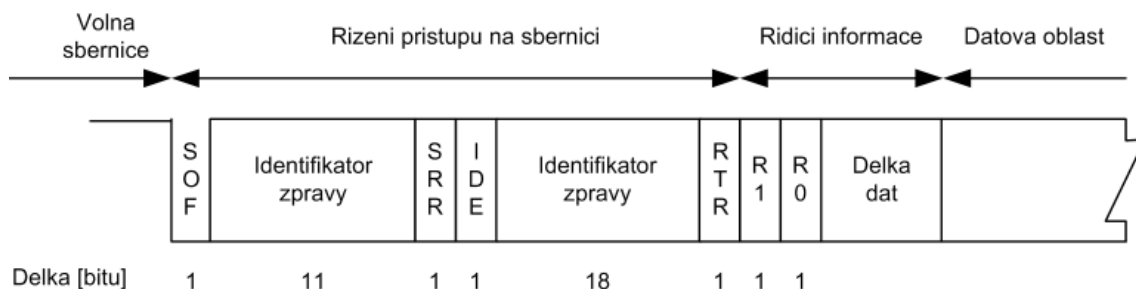


Obrázek 4.4: Uspořádání sběrnice CAN [13]

- **SOF** [*Start of Frame*] (1b) – začátek zprávy
- **Identifikátor zprávy** [*Arbitration frame*] (11b) – nese prioritu a význam zprávy
- **RTR** [*Remote Request*] (1b) – datová zpráva (dominant) x žádost o přístup ke sběrnici (recessive)
- **R0 a R1** [*Control Field*] (2b) – řídicí pole (rezervováno)
- **Délka datové zprávy** (4b)
- **Datová oblast** [*Data Field*] (0 – 64b) – max. 8Byte dat
- **Zabezpečovací kód** [CRC] (15b)
- **CRC oddělovač** [ERC] (1b) – dominant
- **Potvrzení** [ACK] (1b)
- **Oddělovač** [ACD] (1b) – recessive
- **Konec zprávy** [*End of Frame*] (7b) – recessive
- **Mezera mezi zprávami** [*Interframe Space*] (3b) – recessive

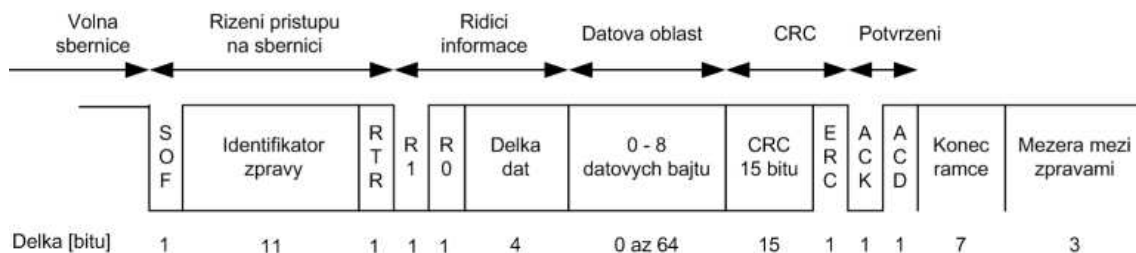
Popis zprávy CAN 2.0B je shodný s předchozí verzí, navíc definuje rozšířený formát zprávy, který je popsán v následujícím odstavci.

Standardní rámec zprávy (*Standard Frame*) má délku identifikátoru 11 bitů a je téměř totožná se specifikací CAN 2.0A, je vidět na brázku 4.5. Rozdílem je pouze bit R1, který zde slouží k určení, zda se jedná o zprávu ve standardním nebo rozšířeném formátu. Dle normy CAN 2.0B je tento bit označován jako IDE (*Identifier Extended*). Pro standardní formát nabývá hodnoty *dominant*, pro rozšířený hodnoty *recessive*. Přístup na sběrnici je v tomto případě dán identifikátorem a bitem RTR (*Remote Request*). [13]



Obrázek 4.5: Uspořádání sběrnice CAN [13]

Rozšířený rámec zprávy (*Extended Frame*) má délku identifikátoru 29 bitů a je zobrazen na obrázku 4.6. Ten je rozdělen na dvě části. O délkách 11 bitů (jako standardní formát) a 18 bitů. Dále je zde zaměněn bit RTR bitem SRR (*Substitute Remote Request*), který vždy nabývá hodnoty *recessive*. Díky tomuto je zabezpečena situace při kolizi identifikátorů standardního a rozšířeného formátu, sběrnici vždy získá zpráva se standardním formátem. Bit IDE (*Identifier Extended*) je vždy nastaven na hodnotu *recessive* a bit RTR je posunut až za konec identifikátoru. Pro vyhodnocení přístupu na sběrnici se postupuje v tomto pořadí dle priority: ID (11bit), SRR, IDE, ID (18 bit), RTR. [13]



Obrázek 4.6: Uspořádání sběrnice CAN [13]

Žádost o data

Tato zpráva má podobný formát jako datová zpráva. Rozdíl je pouze v RTR bitu, který je nastaven na hodnotu *recessive* a zpráva neobsahuje žádnou datovou část. Při žádosti o data se identifikátor zprávy nastaví stejně, jak identifikátor datové zprávy uzlu od kterého data potřebujeme. Díky tomuto, při současném vysílání a žádosti o data má přednost uzel, který data vysílá. Jelikož jeho zpráva má v RTR bitu hodnotu *dominant*, což znamená vyšší prioritu. [13]

Zpráva o chybě

Tato zpráva se používá k indikaci chyb na sběrnici. Pokud je v jakémkoliv uzlu detekována chybná zpráva, která neodpovídá standardům (chyba vkládání bitů, CRC, chyba rámce, chyba bitu), je okamžitě generována zpráva o chybě. Která může mít aktivní (6 bitů *dominant*) nebo pasivní (6 bitů *recessive*) příznak, podle stavu uzlu ve kterém se chyba vyskytla. V případě aktivního příznaku je zpráva poškozena chybou vkládání bitů, v tom případě tuto chybu začnou vysílat i ostatní uzly. Doba vysílání chybových příznaků může být dlouhá 6

až 12 bitů, ne však delší. Po chybovém příznaku každý uzel začne vysílat bity *recessive* a zároveň sledovat zda se na sběrnici neobjeví stejný bit. V případě nalezení prvního bitu *recessive* je odeslán na sběrnici oddělovač chybové zprávy, který obsahuje dalších 7 bitů *recessive*. [13]

Zpráva o přetížení

Touto zprávou je indikováno přetížení uzlu a vytvoření pauzy mezi dalšími zprávami (datovými nebo žádost o data). Toto chování je možné spatřit u zařízení, které jsou zahlceny zprávami a již další nejsou schopny zpracovat. Struktura této zprávy je téměř totožná se zprávou o chybě, ale může být vysílána až po konci aktuální zprávy nebo po oddělovači chyb či po oddělovači předchozího přetížení. [13]

4.1.5 AS-I (*AS-Interface*)

Actuator Sensor Interface (AS-I) se používá v nejnižší úrovni automatizačního provozu. Umožňuje propojení akčních členů, různých senzorů od různých výrobců, které spolu komunikují po jedné sériové lince. Sběrnice *AS-Interface* je typu master–multislave. To znamená, že existuje jedno hlavní zařízení *Master*, ke kterému jsou připojena ostatní zařízení *Slave*. Ve svoji původní specifikaci *AS-Interface 1.0* k ní bylo možno připojit maximálně 31 zařízení. V následující verzi *AS-Interface 2.1* to bylo již 62 zařízení. V roce 2001 byla vytvořena další specifikace, která měla za úkol zvýšení bezpečnosti díky podpoře SIL3. Poslední verze *AS-Interface 3.0* rozšiřuje použití o přenos 16-ti bitových dat, jak ve více, tak i v jediném cyklu.

AS-Interface je kompatibilní pro fieldbus sítě, jako například *Profibus*, *DeviceNet*, *Interbus* a průmyslový Ethernet. K tomuto rozhraní lze připojit mnoho různých zařízení, od PC, senzorů, bezpečnostních modulů až po PLC. *AS-Interface* lze zapojit do jakékoliv topologie (sběrnice, kruh, hvězda) a není ho třeba na koncích ukončovat například odporem. [7]

Charakteristiky AS Interface:

- Nekroucená nestíněná dvojlinka s libovolnou topologií
- Přenos dat a napájení po jednom vodiči
- Délka vedení max. 100m, s opakovači max. 300m
- Komunikace Master/Slave
- Přenos zabezpečen paritou, start a stop bitem
- Pro zabezpečení dat použit kód Manchestero
- Jednoduchá instalace
- Vysoká rychlost komunikace
- Max. 31 slave zařízení ve verzi 1.0 (62 ve verzi 2.1)
- Verze 1.0 -124 binárních vstupů (senzorů) a 124 binárních výstupů (akčních členů)
- Verze 2.1 -248 vstupů a 186 výstupů
- Verze 3.0 -248 vstupů a 248 výstupů

4.1.6 DeviceNet

Používá se na přenos dat na nejnižší úrovni, pro připojení snímačů a akčních členů. *DeviceNet* je otevřená síť přístupná všem různým výrobcům, kteří nemusí nakupovat speciální vybavení ani licenci.

DeviceNet je založen na protokolu CAN (*Controller Area Network*), který byl vyvinut pro automobilový průmysl. Je tedy spolehlivý, odolný vůči rušení a přináší spoustu dalších výhod, jako například dostupnost integrovaných obvodů. *DeviceNet* byl vytvořen společností *Allen-Bradley* a v současnosti je využíván více jak stovkou firem. [7]

Charakteristiky DeviceNet:

- Komunikace Multi-Master, Master/Slave nebo peer to peer s možností Multi-Cast
- Přenos dat rychlostí 125, 250 nebo 500kb/s
- Maximální délka sítě 100 až 500m
- Možnosti napájení připojených členů po sběrnici
- Datové pakety délky 0 až 8 bitů
- Využití standardu CAN na 2. vrstvě ISO/OSI modelu
- Využití standardu CIP na 5. a vyšších vrstvách
- Možnost volby galvanicky odděleného fyzického rozhraní
- Cenově a hardwarově dostupné

4.1.7 InterBus

InterBus byl vyvinut společností *Phoenix Contact*, rozšířil se do mnoho odvětví a je využíván více jak šesti sty firmami. Jedná se o otevřenou průmyslovou sběrnici, pro připojení I/O zařízení k síti. Ke komunikaci je použita nestíněná dvojlinka, po které probíhá sériová komunikace mezi jednotlivými prvky sítě.

Tato sběrnice využívá kruhovou topologii, ke které jsou připojeny jednotlivé aktivní systémy. Základem je hlavní kruh, který lze dále rozšiřovat o nižší úrovně, které mají topologii otevřený strom, ale to však maximálně na 16 úrovní. Díky těmto vlastnostem je tato struktura velmi flexibilní a přizpůsobitelná, můžeme k ní připojit až 512 zařízení se 4096 I/O body. Každé zařízení, které přijme signál, ho zesílí a odešle k sousednímu zařízení. Tímto je zajištěna velká délka sběrnice, která může dosahovat až 13 km. Je zde uplatněna komunikace *Master-Slave*, kde *Master* jsou všechny zařízení na hlavní smyčce a ostatní připojená zařízení se staví do role *Slave*. Není zde třeba zakončovacích odporů, jelikož tuto vlastnost zajišťují všechny poslední zařízení typu *Slave*. [7]

Charakteristiky InterBus:

- Komunikace Master/Slave
- Přenosová rychlost 500kb/s
- Až 512 zařízení typu Slave s 4096 I/O body

- Délka sítě max. 13 km
- Topologie aktivní kruh
- Délka mezi sousedními zařízeními max. 400m (měděné vodiče) a až 3600m (optické vlákno)

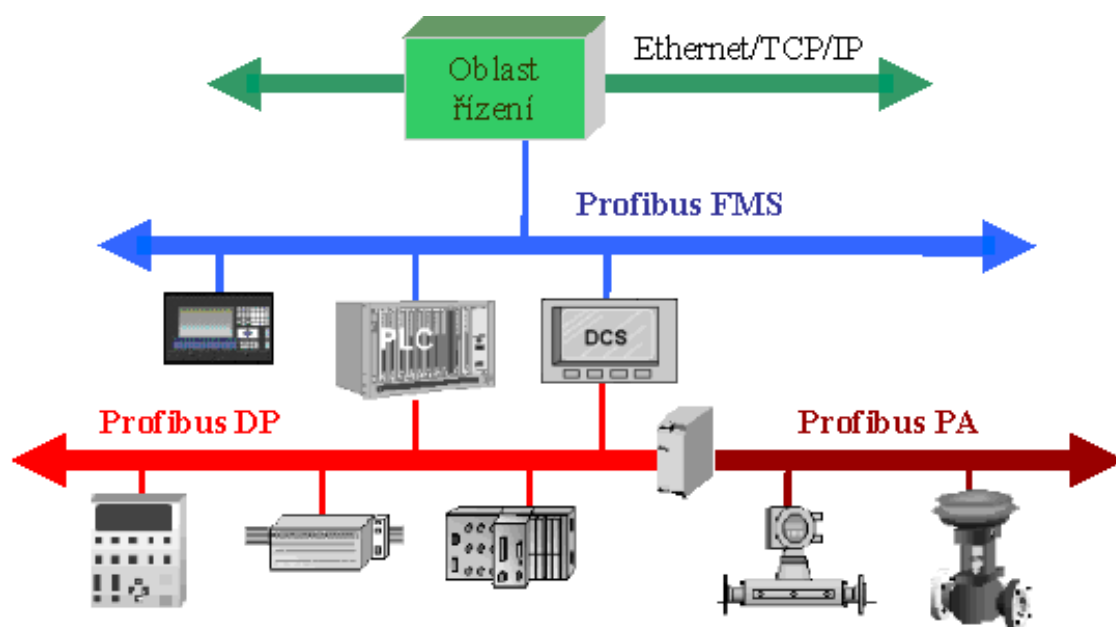
4.1.8 ProfiBus

PROcess Field BUS (*ProfiBus*) je jedna z nejrozšířenějších komunikačních standardů v oblasti průmyslové automatizace. Hlavní výhodou je otevřenost, nezávislost, stabilita, zabezpečená komunikace mezi jednotlivými prvky a bezproblémová interakce mezi zařízeními od různých výrobců. *ProfiBus* je založen na komunikačním modelu ISO/OSI, ze kterého využívá 1., 2. a 7. vrstvu. [7]

- **Fyzická vrstva** – určuje přenosové médium, vysílání a přijímání dat
- **Spojová vrstva** – k přístupu ke sběrnici používá MAC
- **Aplikační vrstva** – propojení sběrnice s procesorem, slouží k aplikačnímu programování

Norma EN 50170 definuje tři komunikační standardy zobrazené na obrázku 4.7. [7]

- **ProfiBus DP** (*Decentralized Periphery*) – nejjednodušší a nejrozšířenější varianta, která je využívána pro rychlý přenos dat mezi I/O jednotkami v běžném průmyslu. Komunikace probíhá pomocí kroucené dvojlinky, nebo optického vlákna, rychlost až 12 Mbit/s.
- **ProfiBus FMS** (*Fieldbus Message System*) – pro komunikaci mezi PLC a field zařízeními. Komunikace v heterogenním prostředí s velkou škálou služeb pro práci s daty, či programy. Médium pro přenos je totožné jako u *ProfiBus DP*.
- **ProfiBus PA** (*Process Automation*) – využívá se v náročných prostředích (výbušné prostředí) v procesní automatizaci pro řízení pomalých procesů. Využívá speciální fyzickou vrstvu, kde komunikace probíhá stálou rychlostí 31,25 kbit/s.



Obrázek 4.7: Standardy ProfiBus [14]

Kapitola 5

Konečný návrh

Tato kapitola je věnována návrhu a koncepci modulárního zobrazovacího systému, jsou zde shrnuty jeho požadované vlastnosti. Také je zde popsána funkčnost jednotlivých modulů a jejich komunikace v celkovém řešení. Nachází se zde i blokové schéma, které je popsáno.

5.1 Vlastnosti

V konečném návrhu modulárního zobrazovacího systému bylo třeba zajistit několik důležitých vlastností, aby celý systém fungoval. Proto jsou jeho hlavní vlastnosti:

- Ovládání pomocí Bluetooth rozhraní
- Modul na bázi LED diod
- Rozšiřitelnost modulů do větších celků
- Komunikace modulů po CAN sběrnici
- Zobrazení libovolného RGB obrazce

Všechny tyto vlastnosti jsou důležité pro funkci celého systému. Díky ovládání pomocí Bluetooth rozhraní se stává interakce uživatele s LED displejem jednodušší. Uživatel si nainstaluje aplikaci určenou pro platformu Android a dále s ní již ovládá celý připojený displej, na kterém zobrazuje informace dle libosti.

Osazení displeje LED diodami je důležité z několika důvodů, které jsou nízká spotřeba elektrické energie, vysoká svítivost, zobrazení všech barev spektra RGB, velký pozorovací úhel. Dále je display možný sestavit do různých tvarů a rozšířit ho do větších celků. V neposlední řadě se o komunikaci mezi moduly stará CAN sběrnice, která disponuje komunikací pomocí jednoho páru vodičů. Tím pádem šetří potřebu mnoha vodičů a složité komunikace mezi jednotlivými moduly.

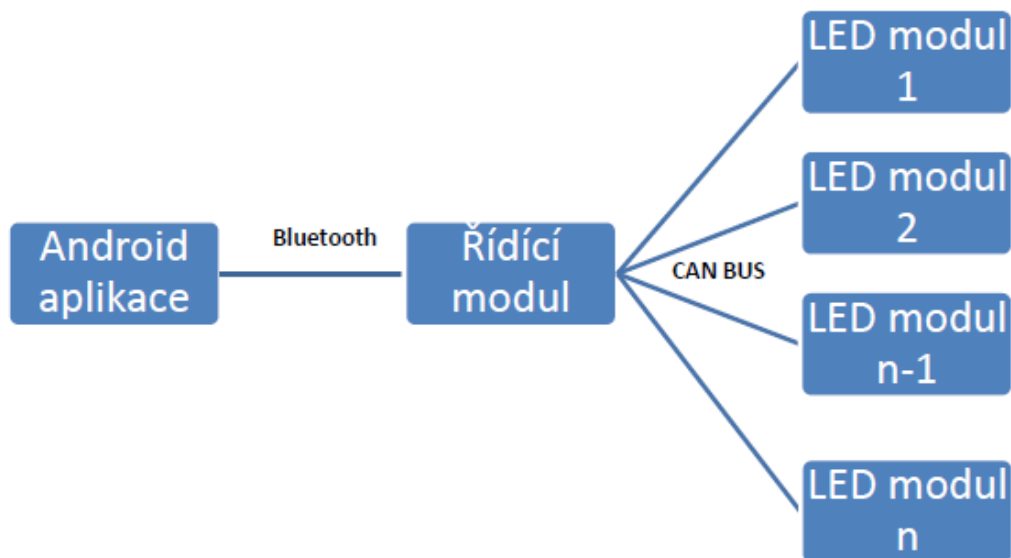
Tyto vlastnosti zaručují použití displeje v mnoha směrech, kde je zapotřebí tento specifický typ displeje. Například pro zobrazení reklam nebo jako velkoplošný LED displej či libovolná informační tabule. Uživatel si může vytvořit displej dle vlastních požadavků a zobrazovat na něm jak jednobarevné, tak i vícebarevné obrazce z pohodlí svého mobilního telefonu.

5.2 Koncepce

Modulární zobrazovací systém je rozdělen do tří samostatných částí, které jsou:

- LED modul
- Řídící modul
- Aplikace Android

Tyto prvky tvoří jeden celek a jeho koncepce je zobrazena na schématu na obrázku 5.1. Je vidět že Android zařízení komunikuje s řídicím modulem pomocí Bluetooth komunikace. Zde dochází k přijímání požadavků od uživatele a jejich odesílání. Řídící modul je složen z vývojové desky *Raspberri pi 3* a komunikačního modulu pro CAN rozhraní a má dostačující výpočetní výkon. Tento prvek zastává hlavní řídicí funkci, a na základě přichozích informací rozhoduje, jaká data budou rozesílána na připojené LED moduly. Komunikace mezi těmito prvky probíhá pomocí CAN rozhraní, kde každý LED modul má svoji vlastní adresu pomocí které je snáze adresovatelný pro řídicí modul. Na posledním místě se nachází již zmíněný LED modul, který je složen z vývojové desky *Arduino Nano*, ke které je připojen CAN BUS modul a maticový RGB LED displej. Tento displej je vyroben firmou *AdaFruit* a nese označení *NeoMatrix*, vyznačuje se rozlišením 8 x 8 RGB LED diod, které obsahují řídicí prvek *WS2812*. Díky tomuto čipu je každá LED dioda adresovatelná samostatně a může být ovládána nezávisle na ostatních. O obsluhu tohoto displeje se stará *Arduino Nano*, které rozhoduje na základě přichozích dat z CAN sběrnice, jaká operace se bude provádět s LED displejem. Důležitou funkcí je možnost spojování LED modulů do větších celků, což znázorňují na obrázku buňky s názvem LED modul. Ty jsou mezi sebou propojeny pomocí jednoho páru vodičů a samotný modul může být dále rozšířen až o další 4, které se dají připojit z každé strany tohoto modulu.



Obrázek 5.1: Koncepce celého systému

Kapitola 6

Hardwarové řešení

Tato kapitola popisuje hardwarový návrh jednotlivých částí zobrazovacího systému. Je zde rozebráno zapojení a zobrazeny schémata. Jednotlivé návrhy desek a schémata byly tvořeny v programu *Eagle 7.6.0*.

6.1 LED modul

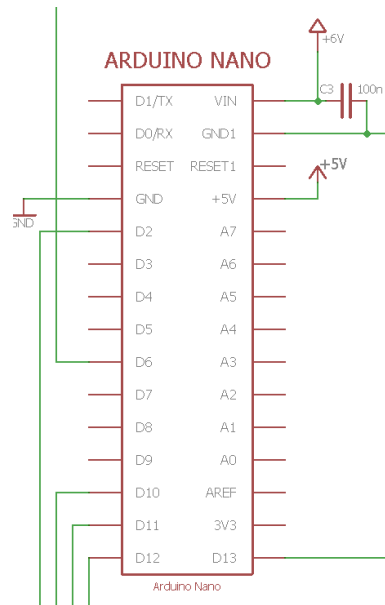
LED modul je složen z ovládacího prvku, CAN BUS modulu, dvou svorek pro připojení LED displeje a čtyř svorek CON1–CON4 pro rozšíření o další LED moduly. Celý modul je napájen napětím 6V.

6.1.1 Ovládací prvek

Jako ovládací prvek je použit vývojový kit Arduino Nano, který je popsán v kapitole [3.1.3](#) a také ho můžeme vidět na obrázku [6.1](#). Tento kit slouží pro ovládání CAN BUS modulu a LED displeje. Kondenzátor C3 slouží k filtraci napájení, které je dále přivedeno na pin Vin. Zde je stabilizováno obvodem 7805 na 5V a zajišťuje napájení vývojového kitu a CAN BUS modulu. Pin D6 slouží k ovládání LED displeje (NeoMatrix 8 x 8) pomocí PWM komunikace. Obsluha CAN BUS modulu probíhá pomocí rozhraní SPI, které se nachází na pinech D10 (CS), D11 (MOSI), D12 (MISO), D13 (SCK) a pinu D2 pro zjištění příchozí zprávy.

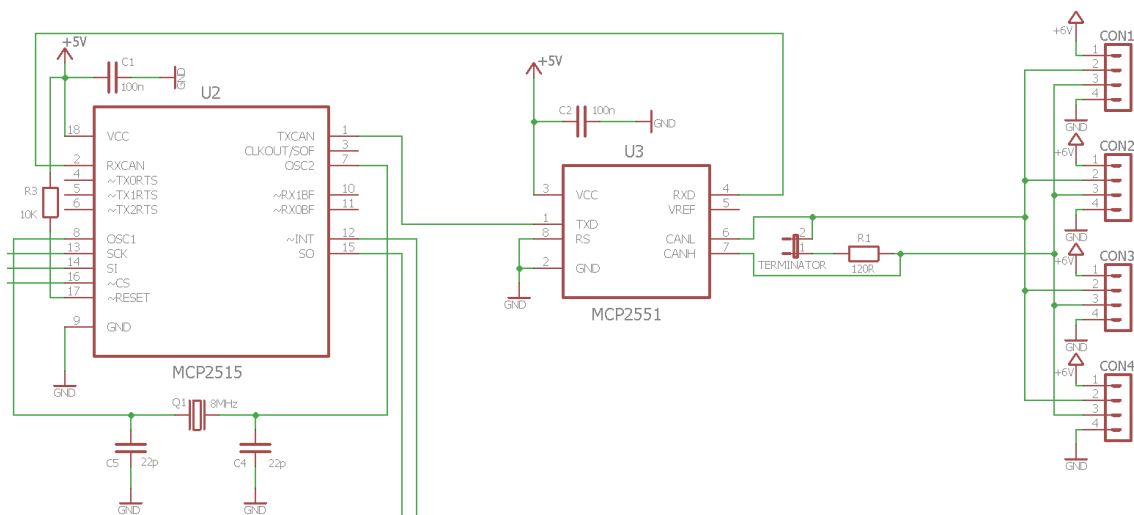
6.1.2 CAN BUS modul

CAN BUS modul, který můžeme vidět na obrázku [6.2](#), umí komunikovat na CAN sběrnici, která je popsána v kapitole [4.1](#). Tento modul je složen z dvojice integrovaných obvodů *MCP2515* a *MCP2551*. První jmenovaný čip je vyroben firmou *Microchip semiconductor*, podporuje specifikaci CAN 2.0B a komunikační rychlost až 1 Mb/s. Podporuje přijímání a odesílání standardních i rozšířených rámců. Lze nastavit dvě masky a až šest filtrů, které pomáhají odfiltrovat nevyžádané zprávy a tím snížit zatížení čipu. *MCP2515* komunikuje s ovládacím prvkem pomocí SPI rozhraní (10MHz), na které jsou odesílá přijatá data ze sběrnice CAN. Toto rozhraní se nachází na pinech 13 (SCK), 14 (MOSI), 15 (MISO) a 16 (CS). Dále tento čip komunikuje pomocí pinů 1 (TXCAX) a 2 (RXCAN) s obvodem *MCP2551*, který funguje jako vysokorychlostní vysílač/přijímač na CAN sběrnici. Tento čip zprostředkovává rozhraní mezi CAN protokolem (*MCP2515*) a fyzickou vrstvou CAN. Podporuje rychlost až 1 Mb/s a normu ISO 11898-2, která určuje požadavky na fyzickou



Obrázek 6.1: Schéma ovládacího prvku

vrstvu. Data jsou přijímána z CAN sběrnice pomocí pinů 6 (CANL) a 7 (CANH) a distribuována na CAN kontrolér pomocí pinů 1(TXD), 4(RXD). Odpor R1 o velikosti $120\ \Omega$ je použit jako zakončovací odpor CAN sběrnice a je možné ho sepnout pomocí jumperu *Terminator*. Signály CAN L a CAN H jsou dále vedeny na svorky CON1–CON4 a šířeny na okolní LED moduly. Kondenzátory C1 a C2 jsou zde použity k filtraci napájení. Krystal Q1 s kondenzátory C4 a C5 je využit jako zdroj hodin pro integrovaný obvod *MCP2515*.



Obrázek 6.2: Schéma CAN BUS modulu

Kapitola 7

Programové řešení

7.1 LED modul

Tento modul slouží k přijímání dat z CAN sběrnice a na jejich základě k ovládání LED displej.

7.1.1 Arduino

Jelikož je LED modul založen na modulu *Arduino Nano*, program je napsán v programu *Arduino IDE* v programovacím jazyku *Wiring*. Více o tomto jazyku se můžeme dozvědět v kapitole 3.1.2. Program je rozdělen do 2 hlavních částí. První funkce `void setup()` je provedena pouze jednou, probíhá zde inicializace CAN rozhraní a LED displeje. V další funkci `void loop()` se nachází switch, který pomocí příchozích dat rozhoduje, jaká akce se má provést s LED displejem..

7.1.2 CAN BUS rozhraní

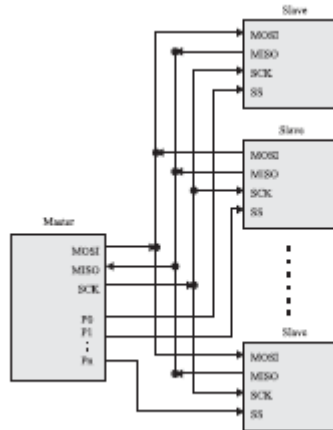
Toto rozhraní slouží ke komunikaci na CAN sběrnici a je tvořeno dvojicí integrovaných obvodů MCP2515 a MCP2551. Ke komunikaci s mikroprocesorem slouží první jmenovaný čip, kde následná komunikace probíhá přes rozhraní SPI.

Rozhraní SPI

SPI (*Serial Peripheral Interface*) je sériové periferní rozhraní. Používá se pro komunikaci mikroprocesoru s připojenými perifériemi (displeje, A/D převodníky, bluetooth a další). Základní skladbu tohoto rozhraní můžeme vidět na obrázku 7.1. Podmínkou fungování je mít dvě a více zařízení, kde vždy jedno musí být řídicí obvod (mikroprocesor) typu *Master* a ostatní připojené periférie typu *Slave*. Ke komunikaci je využito čtyř vodičů. [5]

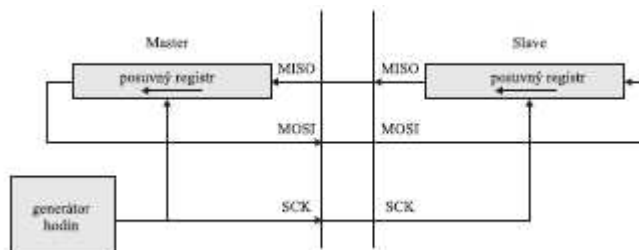
- **MOSI** (*Master Out, Slave In*) – datový výstup zařízení *Master*, slouží k odesílání dat na periférie typu *Slave*.
- **MISO** (*Master In, Slave Out*) – datový vstup zařízení *Master*, slouží k příjmu dat z periférií typu *Slave*.
- **SCK** – výstup hodinového signálu zařízení *Master*, který je šířen dále na piny SCK všech připojených periférií.

- **CS/SS** (*Slave Select*) – slouží pro výběr *Slave* zařízení, se kterým bude vedena komunikace. Vstupy SS jsou k *Master* zařízení jsou připojeny pomocí samostatných vodičů, díky tomu lze jednoduše určit obvod, se kterým bude zahájena komunikace.



Obrázek 7.1: Schéma SPI [5]

Komunikace vždy probíhá pouze mezi zařízením *Master* a libovolnou periferií *Slave*. K přenosu dat jsou využity posuvné registry obou zařízení, které jsou v okamžiku komunikace připojeny. Toto propojení je zobrazeno na obrázku 7.2. [5]

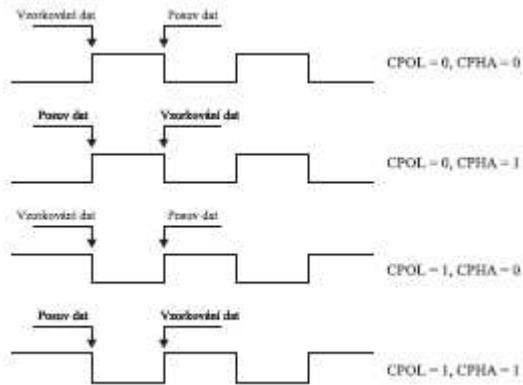


Obrázek 7.2: Posuvné registry SPI [5]

Posouvání jednotlivých registrů je řízeno pomocí hodinového signálu, který vysílá zařízení *Master*. Vztahy mezi hodinovým a datovým signálem je dán parametry CPOL a CPHA, které se dají nastavit v řadiči a jsou na obrázku 7.3. Maximální velikost hodinového signálu, se kterým dokáže rozhraní SPI pracovat je 2 MHz a napěťová úroveň je dána použitou technologií. [5]

Komunikace s CAN BUS modulem

Inicializace je provedena funkcí `CAN0.Begin(MCP_STDEXT, CAN_500KBPS, MCP_8MHZ)`. První parametr je určen k nastavení módu komunikace, zda bude prvek přijímat i odesílat data na sběrnici, v tomto případě znamená, že bude použito filtrování příchozích zpráv. Dalším parametrem je nastavení komunikační rychlosti, která je v našem případě nastavena na 500Kb/s. Posledním parametrem se nastavuje rychlost hodin, které jsou připojeny k čipu MCP2515. Dále je port D2 nastaven jako vstupní a na jeho základě se indikují nově příchozí



Obrázek 7.3: Posuvné registry SPI [5]

zprávy z CAN rozhraní, které jsou následně přijímány funkcí `CAN0.readMsgBuf(&rxId, &len, rxBuf)`. Zde první parametr slouží k přenosu ID příjemce, další jako délka dat a poslední obsahuje přenášená data. Také je zde využito nastavení dvou masek pomocí funkce `CAN0.init_Mask(0,0,0x010F0000)` a jednoho filtru `CAN0.init_Filt(0,0,0x01030000)`, který pomáhá přijímat pouze zprávy, které jsou určeny pro tento modul na základě shodné adresy.

Pro obsluhu CAN rozhraní byla využita knihovna `mcp_can`[8], jež vytvořil *Cory J. Fowler*. Tato knihovna je opensource software, který lze dále šířit a upravovat na základě licence *GNU Lesser General Public*.

7.1.3 Ovládání displeje

Displej slouží k vizuálnímu zobrazování dat, které přicházejí ze sběrnice CAN a je tvořen modulem *NeoMatrix* od společnosti *AdaFruit*. Komunikace s tímto modulem probíhá po jednom vodiči a má danou svoji specifikaci.

Nejprve je vytvořena proměnná `matrix` pomocí konstrukturu:

```
Adafruit_NeoMatrix(8, 8, PIN, NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
NEO_MATRIX_COLUMNS + NEO_MATRIX_PROGRESSIVE, NEO_GRB + NEO_KHZ800)
```

Zde se nastavuje počet LED diod na X a Y souřadnici a komunikační port, dalšími parametry je nastaven počáteční bod na displeji, následně jsou zde 2 proměnné na určení směru zobrazování dat a nakonec je nastaveno barevné schéma a rychlost komunikace. Poté následuje spuštění komunikace mezi displejem a mikroprocesorem. Po provedení inicializace je jas displeje nastaven na 40% funkcí `matrix.setBrightness(40)` a pomocí další funkce `matrix.show()` překreslen. V hlavní programové smyčce je displej ovládán v závislosti na příchozích datech, která rozhodují, jaká operace bude provedena.

- Nastavení jasu
- Smazání displeje
- Překreslení displeje
- Vykreslení jednoho bodu

K obsluze LED displeje byla použita knihovna *Adafruit_NeoMatrix*^[3]. Jedná se o open-source software, který je opatřen licenci *GNU Lesser General Public*. Knihovnu je možné kopírovat či upravovat v rámci mezí licence.

7.2 Řídící modul

Tento modul je hlavním srdcem celého systému. Stará se o komunikaci s *Android* zařízením, které komunikuje pomocí Bluetooth rozhraní a na základě jeho požadavků odesílá potřebná data na CAN sběrnici a pomocí nich ovládá LED displej. Modul je naprogramován v jazyce *Python3*. Řídící a CAN BUS modul komunikují pomocí rozhraní SPI, které je popsáno v kapitole 7.1.2.

7.2.1 Hlavní programová smyčka

Kód řídicího modulu je rozdělen do dvou programových celků. V první části, která se provede pouze jednou, je inicializováno Bluetooth a CAN rozhraní. Druhá část je opakována v nekonečném cyklu, zde se v každém opakování čeká na zprávu z Bluetooth rozhraní. Pokud je zpráva přijata, dochází k dekodování, kde první její znak znamená, co se má udát s LED displejem a ve zbytku zprávy se nachází data. Zprávy jsou pomocí soustavy podmínek rozlišeny a následně se generují odpovídající data, která se odesílají na CAN sběrnici:

- Vykreslení barevného textu
- Vykreslení dvoubarevného obrázku
- Nastavení jasu
- Smazání displeje
- Překreslení displeje

Tyto data mají daného příjemce(LED modul), který provede odpovídající akci. V případě *Vykreslení barevného textu* nebo *Vykreslení dvoubarevného obrázku* jsou na CAN sběrnici odesílána data po jednotlivých bodech, které obsahují příjemce, místo na displeji a barvu tohoto bodu. Při *Nastavení jasu* je odeslán identifikátor a číselná hodnota jasu od 0 do 100. V ostatních případech je odeslán jen určitý identifikátor. Po úspěšném odeslání zprávy se opět opakuje nekonečný cyklus.

7.2.2 Bluetooth rozhraní

Nejprve je vytvořen server pomocí `socket(socket.AF_BLUETOOTH, socket.SOCK_STREAM, socket.BTPROTO_RFCOMM)`, tomu je dále přidělena MAC adresa a port pomocí funkce `server.bind(MACAddress, port)`. A funkce `server.listen(backlog)` určuje současný počet připojených zařízení. Po úspěšném vytvoření serveru je na řadě připojení klienta, o to se stará funkce `server.accept()`, kde její návratová hodnota obsahuje adresu připojeného klienta. Pokud i tato operace proběhne úspěšně, nastává přijímání dat od klienta pomocí funkce `client.recv(size)`. Zde proměnná `size` určuje velikost přijímaných dat. Ty jsou dále zpracována a na jejich základě je odeslána určitá informace na CAN rozhraní. Při odpojení klienta je provedena funkce `client.close()`, pomocí které se ukončí vzájemná komunikace. A následně se čeká na připojení nového klienta.

7.2.3 CAN BUS rozhraní

Toto rozhraní je zapnuto pomocí systémového příkazu `os.system("sudo /sbin/ip link set can0 up")`. Následně probíhá jeho inicializace v rámci programu pomocí funkce `can.interface.Bus(channel='can0', bustype='socketcan_native')`, ve které se nastavuje použitá sběrnice a její typ. V případě úspěchu je toto rozhraní připravené a lze s ním pracovat. Důležitou vlastností je posílání dat na tuto sběrnici, které je realizováno pomocí dvou funkcí. V první je vytvořena zpráva pomocí

```
msg = can.Message(arbitration_id=0x100, data="xxx", extended_id=false)
```

Zde první proměnná definuje adresu příjemce, další udává odesílaná data a poslední rozhoduje zda se jedná o standardní nebo rozšířený formát zprávy. Druhá funkce `can.send(msg)` zajišťuje následné odesílání této zprávy. Po ukončení programu je rozhraní opět vypnuto pomocí systémového příkazu `os.system("sudo /sbin/ip link set can0 up")`.

7.3 Aplikace Android

Jedná se o uživatelskou aplikaci naprogramovanou pro platformu *Android*. Jejím úkolem je zajistit interakci mezi uživatelem a LED displejem. Uživatel si zde volí, k jakému displeji se chce připojit. Po úspěšném připojení již může tento displej ovládat a nastavovat jeho parametry. Aplikace je navrhována a naprogramována ve vývojovém prostředí *Android Studio*. Programovací jazyk je *Java*.

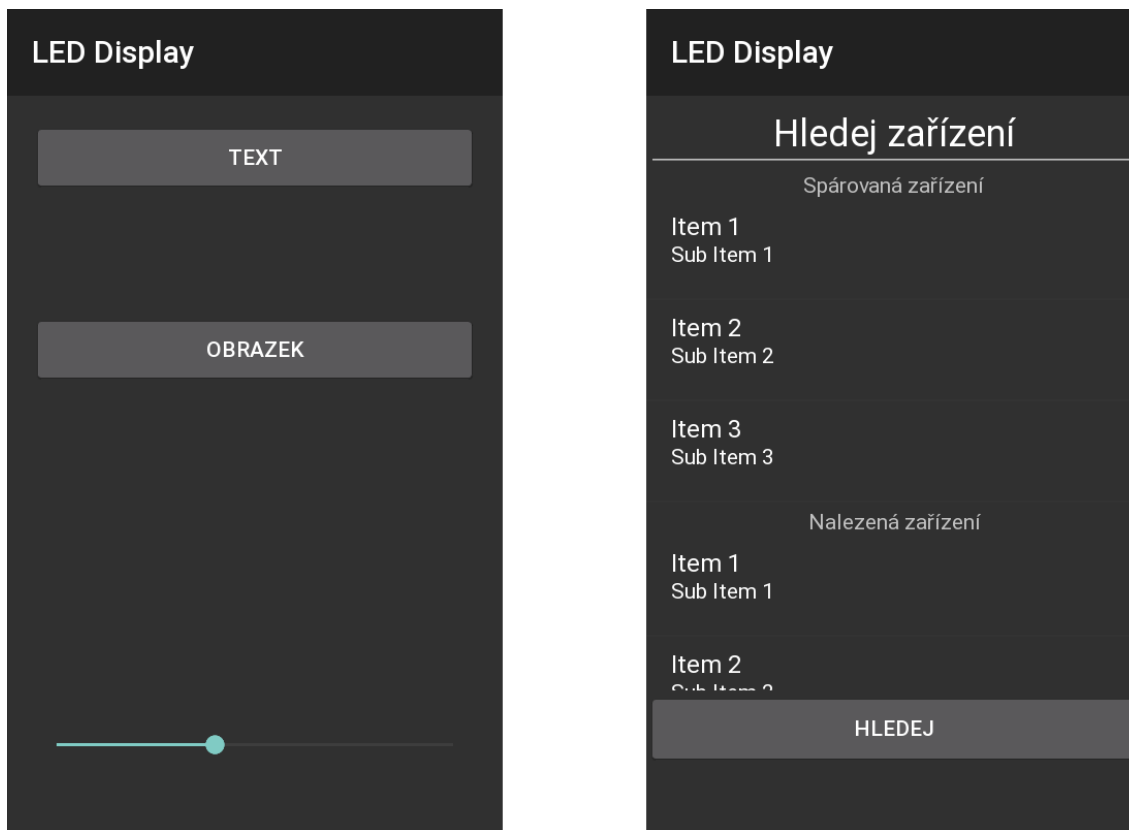
7.3.1 Hlavní aktivita

Tato aktivita slouží jako hlavní zobrazovací okno, které se zobrazí při spuštění aplikace a zastává tuto funkci po celou dobu běhu programu. Z hlavní aktivity je dále možno otevírat pomocné dialogy, které zastávají různé funkce. Toto okno je zobrazeno na obrázku 7.4 vlevo. Je zde vidět že se jedná o zcela jednoduché a srozumitelné uživatelské rozhraní, které zvládne ovládat i začátečník. Jeho hlavním základem jsou dvě tlačítka, pomocí kterých se odesílá informace o tom, co se má zobrazit na připojeném displeji. Jedná se o tlačítko pro zobrazení dvoubarevného obrázku a druhé pro zobrazení barevného textu. Dále se zde nachází komponenta *SeekBar*, která slouží pro nastavování jasu celého displeje. Nedílnou součástí je také menu, které se nachází v pravém horním rohu a obsahuje tři položky. Po kliknutí na tyto položky jsou otevřeny odpovídající dialogy, které jsou popsány v následujících kapitolách.

7.3.2 Připojení

Dialog *Připojení* slouží k navázání spojení s řídicím modulem pomocí Bluetooth rozhraní. Je spuštěn po kliknutí na ikonu lupy, která se nachází jako první položka v menu. Dialog je rozdělen na dvě zobrazovací části a tlačítko pro zahájení hledání, toto je zobrazeno v grafickém návrhu na obrázku 7.4 vpravo. V první části se nachází již spárovaná zařízení. Tato část je tvořena komponentou *TextView* s textem „Spárovaná zařízení“ a seznamem *ListView* ve kterém jsou zobrazeny jména a adresy zařízení, která jsou již spárovaná. Po kliknutí na vybranou položku ze seznamu je zařízení uloženo, dialog ukončen a probíhá již komunikace v hlavní aktivitě. Druhá část slouží ke zobrazení aktuálně dostupných zařízení, ke kterým se lze připojit. Má totožné vlastnosti jak již zmíněná první část. Tlačítkem ve

spodní části je možné spustit nebo ukončit hledání Bluetooth zařízení. Ty jsou po úspěšném nalezení zobrazeny v seznamu nalezených zařízení.



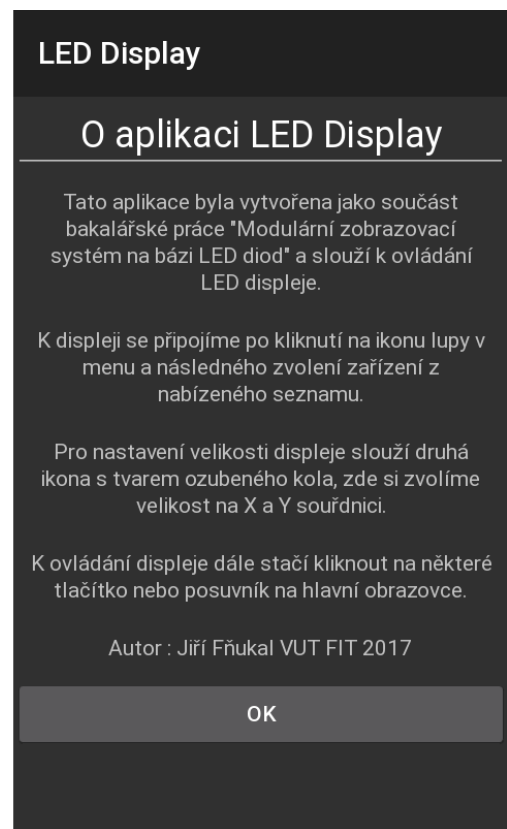
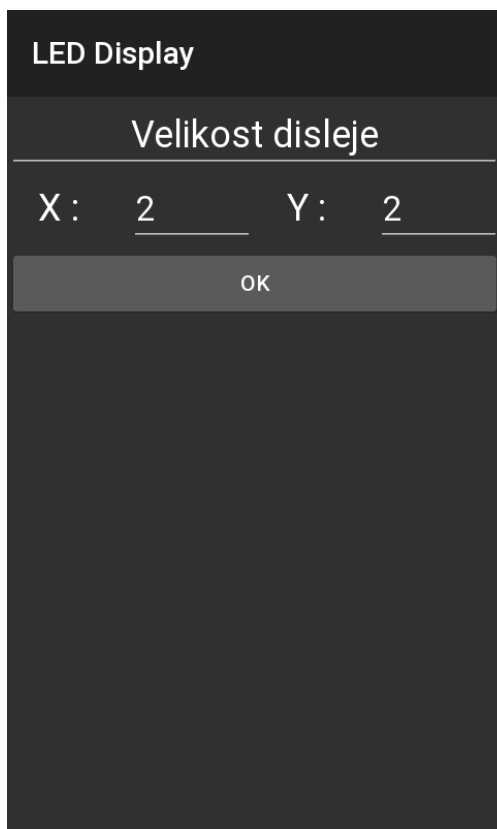
Obrázek 7.4: Vlevo okno Hlavní aktivita, vpravo dialog Připojení

7.3.3 Nastavení

Jedná se o další dialog, který je možno spustit z hlavní aktivity pomocí ikony ozubeného kola. Jeho hlavní funkcí je nastavení parametrů displeje, ke kterému jsme připojeni. Je zde možno nastavit velikost displeje, který se zadává do políček a určuje nám počet modulů na X a Y souřadnici. Po stisknutí tlačítka OK se daná informace odešle přes Bluetooth rozhraní na řídicí modul a je opět zobrazena hlavní aktivita. Jeho návrh je vidět na obrázku 7.5 vlevo.

7.3.4 O aplikaci

Dialog, který je znázorněn na obrázku 7.5 vpravo, slouží pouze k popsání aplikace. Je zde napsán její autor a uživatel se zde může dozvědět jak aplikaci ovládat. Po kliknutí na tlačítko OK je dialog ukončen a přechází se opět na hlavní aktivitu.



Obrázek 7.5: Vlevo dialog Nastavení, vpravo dialog O aplikaci

Kapitola 8

Testování

Testování je důležitou činností při tvorbě výrobku. Je zapotřebí, aby byly testy prováděny při každé softwarové nebo hardwarové změně. Jinak by mohlo dojít k chybě, která by mohla být následně těžko dohádatelná a mohla by mít za následek zpomalený vývoj.

8.1 Testování modulárního zobrazovacího systému

Testování probíhalo v několika fázích. V první se testovalo CAN rozhraní pomocí dvou vývojových desek *Arduino* a CAN BUS modulů. Po úspěšné komunikaci přišla na řadu komunikace s Bluetooth modulem a LED displejem, která probíhala v rámci jedné platformy.

V druhé fázi testování již bylo použito nepájivé pole, na kterém byly sestaveny dva moduly CAN BUS. Které byly vytvořeny z integrovaných obvodů a pasivních součástek. Zde byl největší problém s propojovacími cestami, které byly tvořeny izolovanými vodiči, ve kterých se při diagnostice chyb špatně orientovalo. Ale i přes tyto obtíže se podařilo komunikaci na těchto prototypch realizovat. V této fázi testování byla také vytvořena základní aplikace pro komunikaci mezi mobilním zařízením a platformou *Arduino*.

V pořadí třetí fáze testování byla založena na již hotových deskách plošných spojů, které se musely po napájení součástek důkladně zkontrolovat, aby nevznikal zkrat mezi jednotlivými cestami na desce. Po překontrolování se hotový LED modul osadil integrovanými obvody a displejem a proběhlo jeho oživení. Zde proběhlo vše v pořádku, jen se vyskytla menší změna v návrhu napájení desky, která byla v příštích verzích odstraněna. V této části byla zprovozněna základní komunikace mezi všemi moduly a bylo již možné jednoduše ovládat LED displej.

Po několika testech přišla na řadu další fáze, ve které byl řídicí modul do této doby *Arduino Mega* vyměněn za výkonnější *Raspberry Pi 3*, který svým výkonem již dostačoval. To sebou přineslo nemalý problém v nutnosti přepsat zdrojový kód řídicího modulu do jazyka *Python* a dále upravit komunikace přes *Bluetooth* rozhraní, jelikož *Raspberri Pi 3* má již Bluetooth integrované na desce. V této části také byla vytvořena deska plošných spojů pro CAN rozhraní, která měla rozdílnou koncepci v napájení oproti platformě *Arduino*. Po těchto všech úpravách byla provedena konečná série testů, která neodhalila žádné problémy a potvrdila správnou funkci systému.

Aplikace pro Android byla tetována na *Android verzi 4.1.2*. Řídicí modul při testování pracoval na operačním systému *Raspbian verze 4.4*. A LED modul byl testován s jeho základní deskou *Arduino Nano*.

8.2 Možnosti rozšíření

K systému je možno připojit velký počet jednotlivých LED modulů. Hlavním omezením je zde sběrnice CAN, která má maximální přenosovou rychlost dosahující 1 Mb/s. A při připojení velkého množství LED modulů by tato rychlost nemusela dostačovat. Možným řešením by mohla být změna této komunikace za rychlejší, například Ethernet. Nebo použít více jednotlivých řídicích modulů, které by řídily určitý počet LED modulů, a celý tento celek by byl řízen centrálním prvkem s největším výkonem.

Dalším možným omezením by mohl být ovládací prvek LED modulu (*Arduino Nano*), který by při velkých obnovovacích rychlostech nemusel odpovídat požadavkům. Zde by bylo jediným řešením výměna této desky za výkonnější.

Možným vylepšením by mohla být výměna nebo úprava stávajícího zobrazovacího displeje, který by zaručil rychlejší odezvu. Úprava by spočívala v ovládní jednotlivých řádků nebo sloupců displeje samostatně po osmi diodách na rozdíl od stávajících 64.

V případě vylepšení o Ethernetové rozhraní by bylo vhodné také vytvořit aplikaci pro počítač, která by řídila celý celek. Dále by bylo vhodné zajistit podporu pro přehrávání videa či použití displeje jako monitoru k počítači.

Posledním vylepšením by byla úprava stávajícího napájení podle individuálních potřeb instalace. Například pro automobil využitím stabilizátoru napětí či jiného řešení.

Kapitola 9

Závěr

Cílem bakalářské práce bylo sestavit modulární zobrazovací systém na bázi LED diod. Kde jednotlivé moduly je možno zapojit do větších celku, a které mezi sebou komunikují pomocí CAN rozhraní. Dalším požadavkem byla možnost řízení zobrazovacího systému pomocí mobilního telefonu po Bluetooth rozhraní. Tyto požadavky byly splněny a bližší informace o sestavě a vlastnostech celého systému jsou popsány v kapitole 5.1.

Zobrazovací systém je ovládán pomocí aplikace, která je určena pro mobilní platformu Android. Tato aplikace je přehledná a má intuitivní ovládání, které zvládne i méně zkušený uživatel.

Za účelem testování byl vytvořen zobrazovací systém se čtyřmi moduly, které lze libovolně ovládat a rozšiřovat. Jednotlivé moduly jsou rozmístěny do čtverce, který má rozlišení 16x16 RGB bodů. Tento displej může také sloužit například k prezentaci samotného výrobku.

Na základě testování a výsledné konstrukce bych tento zobrazovací systém doporučil používat pro menší zobrazovací celky, jako jsou například informační tabule. A v případě širšího spektra nasazení bych do budoucna navrhnul jisté úpravy, které by vedly ke zrychlení celého systému, ty jsou popsány v kapitole 8.2.

Díky této práci jsem získal mnoho nových zkušeností s návrhem a tvorbou hardware. Také jsem se seznámil s LED zobrazovacím systémy a dozvěděl se mnoho důležitých věcí o sběrnici CAN a vývojových platformách *Arduino* a *Raspberrí Pi*.

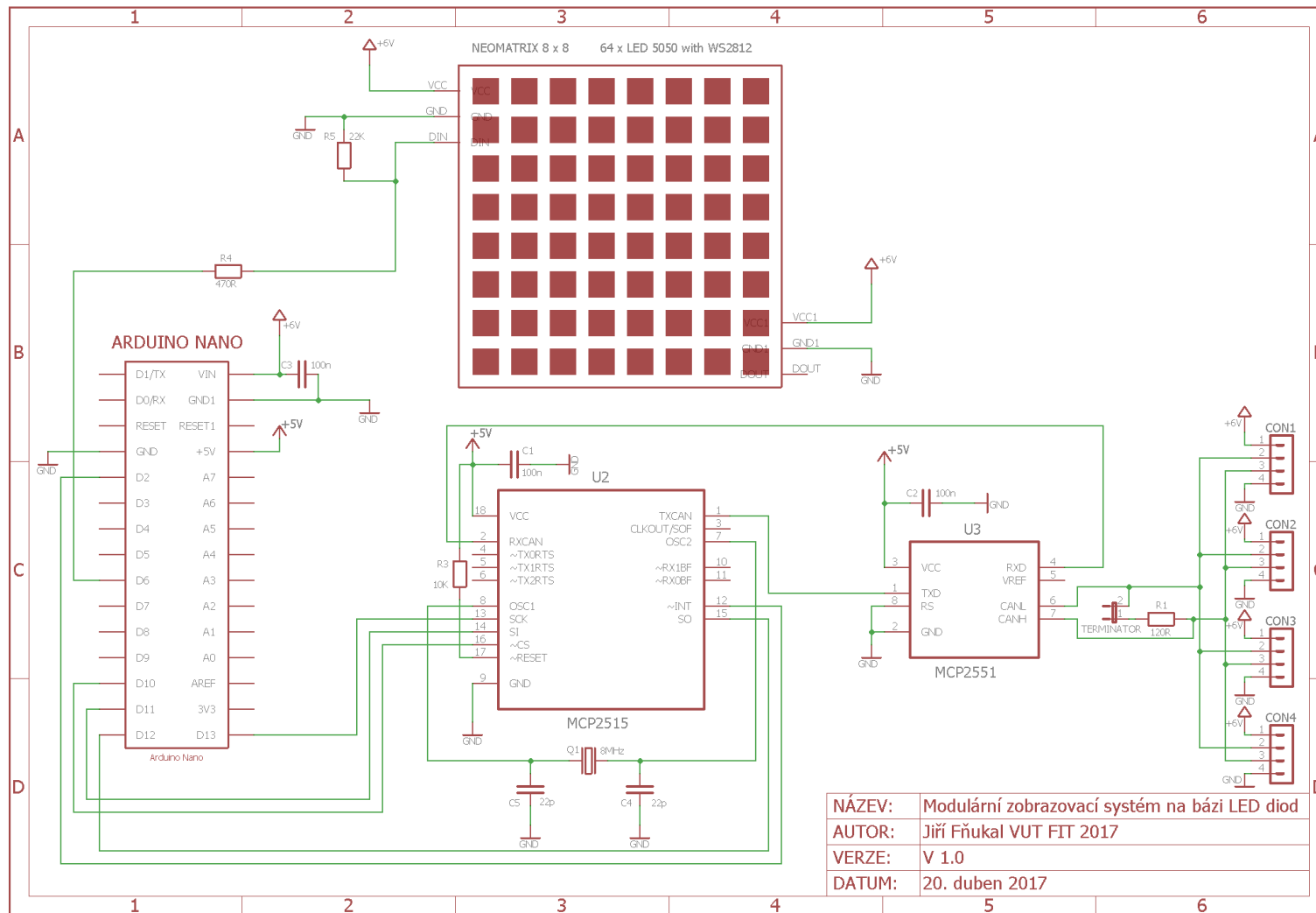
Literatura

- [1] ADJ: *AV6*. ADJ Products,LLC., 2017, [Online; navštíveno 12.05.2017].
URL <http://www.adj.com/av6>
- [2] BQ: *NANO PINOUT*. Srpen 2014, [Online; navštíveno 12.05.2017].
URL <http://www.pighixxx.com/test/pinouts/boards/nano.pdf>
- [3] Burgess, P.: *NeoMatrix Library*. Adafruit, Únor 2016, [Online; navštíveno 12.05.2017].
URL <https://learn.adafruit.com/adafruit-neopixel-uberguide/neomatrix-library>
- [4] CHAUVET: *F4*. CHAUVET Professional, 2017, [Online; navštíveno 12.05.2017].
URL <http://www.chauvetvideo.com/f4>
- [5] Dudáček, K.: *Sériová rozhraní SPI, Microwire, I2C a CAN*. 2002, [Online; navštíveno 12.05.2017].
URL http://home.zcu.cz/~dudacek/NMS/Seriova_rozhrani.pdf
- [6] Elation: *EZ4*. Elation Professional, 2017, [Online; navštíveno 12.05.2017].
URL <http://www.elationlighting.com/ez4>
- [7] Elektroprůmysl.cz: *Průmyslové komunikační sítě*. Elektroprůmysl.cz, Leden 2003, [Online; navštíveno 12.05.2017].
URL <http://www.elektroprumysl.cz/automatizace/prumyslove-komunikacni-site>
- [8] Fowler, C. J.: *MCP_CAN Library*. Červen 2016, [Online; navštíveno 12.05.2017].
URL https://github.com/coryjfowler/MCP_CAN_lib
- [9] G-LEC: *Phantom 30*. G-LEC Europe GmbH, 2008, [Online; navštíveno 12.05.2017].
URL <http://www.promusic.cz/soubory/File/Downloads/Data%20sheet/G-LEC/phantom30.pdf>
- [10] Microsoft: *Raspberry Pi 2 and 3 Pin Mappings*. Microsoft, 2017, [Online; navštíveno 12.05.2017].
URL <https://developer.microsoft.com/en-us/windows/iot/docs/pinmappingsrpi>
- [11] PixelFLEX: *FLEXStorm*. PixelFLEX, Leden 2017, [Online; navštíveno 12.05.2017].
URL <http://pixelflexled.com/wp-content/uploads/2017/02/Pixelflex-FLEXStorm-021017.pdf>

- [12] Planar: *Leyard NVH08S*. Planar Systems, Inc., Květen 2017, [Online; navštíveno 12.05.2017].
URL <http://www.planar.com/products/led-video-walls/nvh/leyard-nv8/?f=Leyard%20NV8.pdf>
- [13] Polák, K.: *Sběrnice CAN*. Červen 2003, [Online; navštíveno 12.05.2017].
URL <http://www.elektrorevue.cz/clanky/03021/index.html>
- [14] Smutný, L.; Klečka, R.: *Průmyslové sítě*. Duben 2005, [Online; navštíveno 12.05.2017].
URL http://www.e-automatizace.cz/ebooks/ridici_systemy_akcni_cleny/S_Profibus.html
- [15] Smutný, L.; Klečka, R.: *Rozdělení průmyslových sítí*. Duben 2005, [Online; navštíveno 12.05.2017].
URL http://www.e-automatizace.cz/ebooks/ridici_systemy_akcni_cleny/S_Rozdeleni.html
- [16] Voda, Z.: *Průvodce světem Arduina*. Nakladatelství Martin Stríž, Bučovice, 2015, ISBN 978-80-87106-90-7.
- [17] Wikipedia: *Raspberry Pi*. Wikimedia Foundation, Duben 2017, [Online; navštíveno 12.05.2017].
URL https://cs.wikipedia.org/wiki/Raspberry_Pi

Příloha A

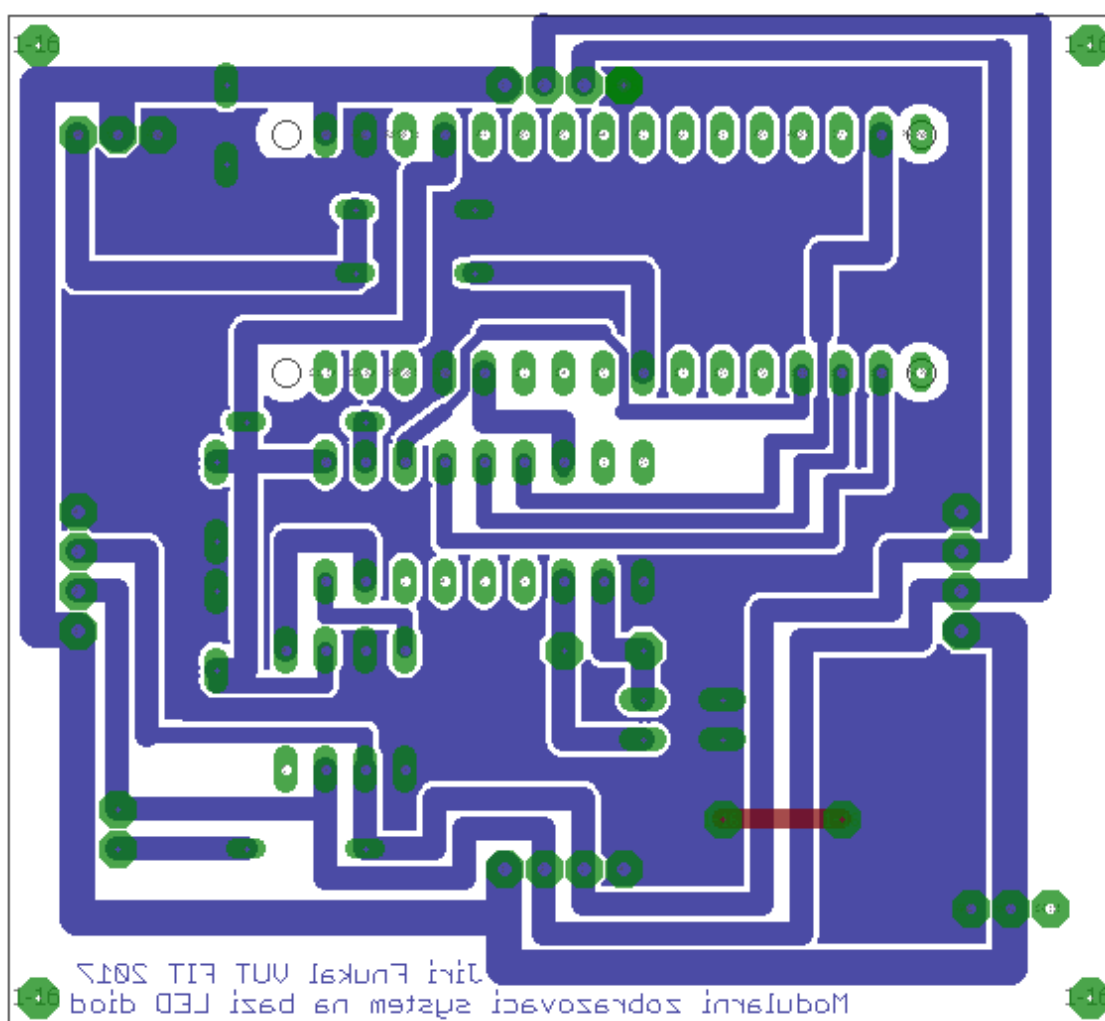
Schéma zapojení LED modulu



Obrázek A.1: Schéma LED modulu

Příloha B

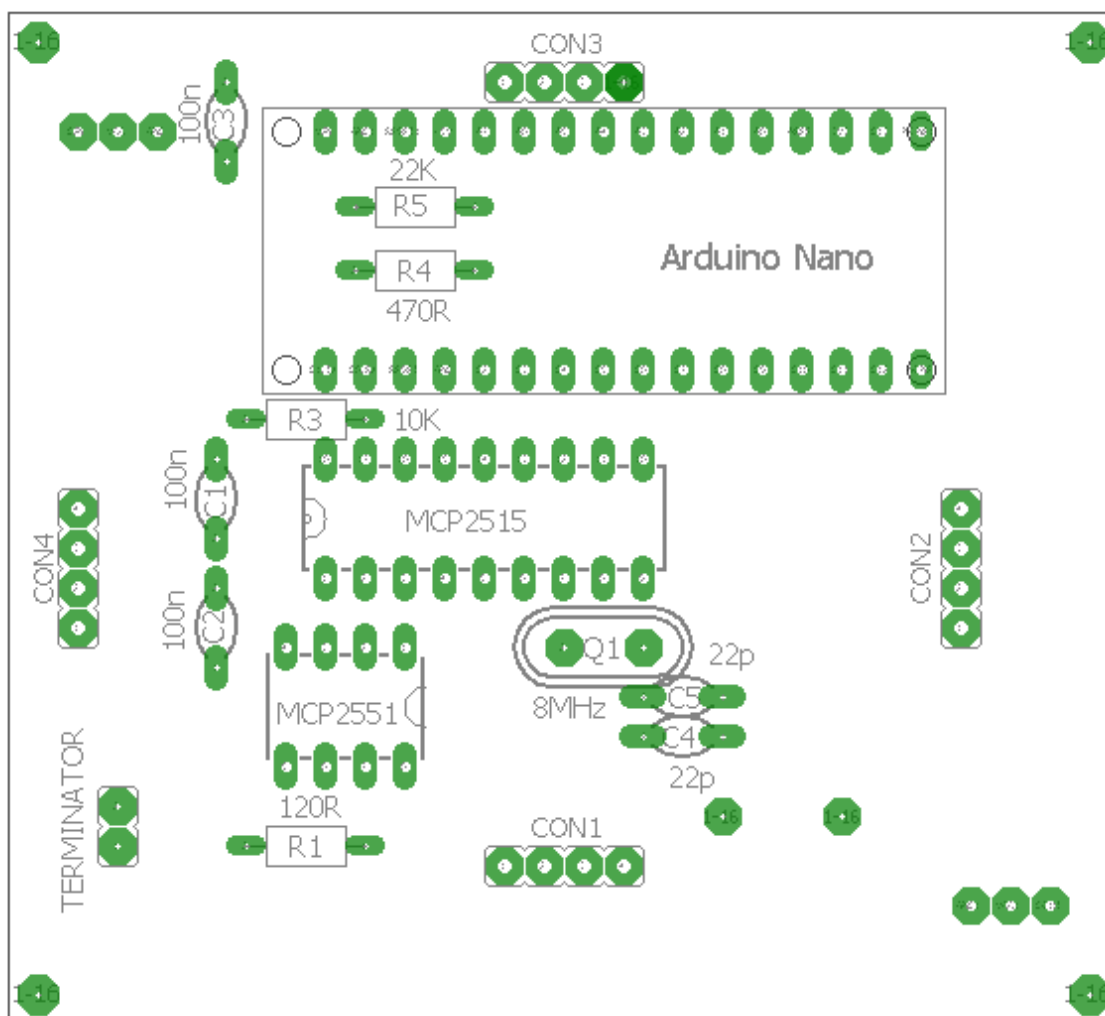
Deska plošných spojů – spoje



Obrázek B.1: DPS spoje

Příloha C

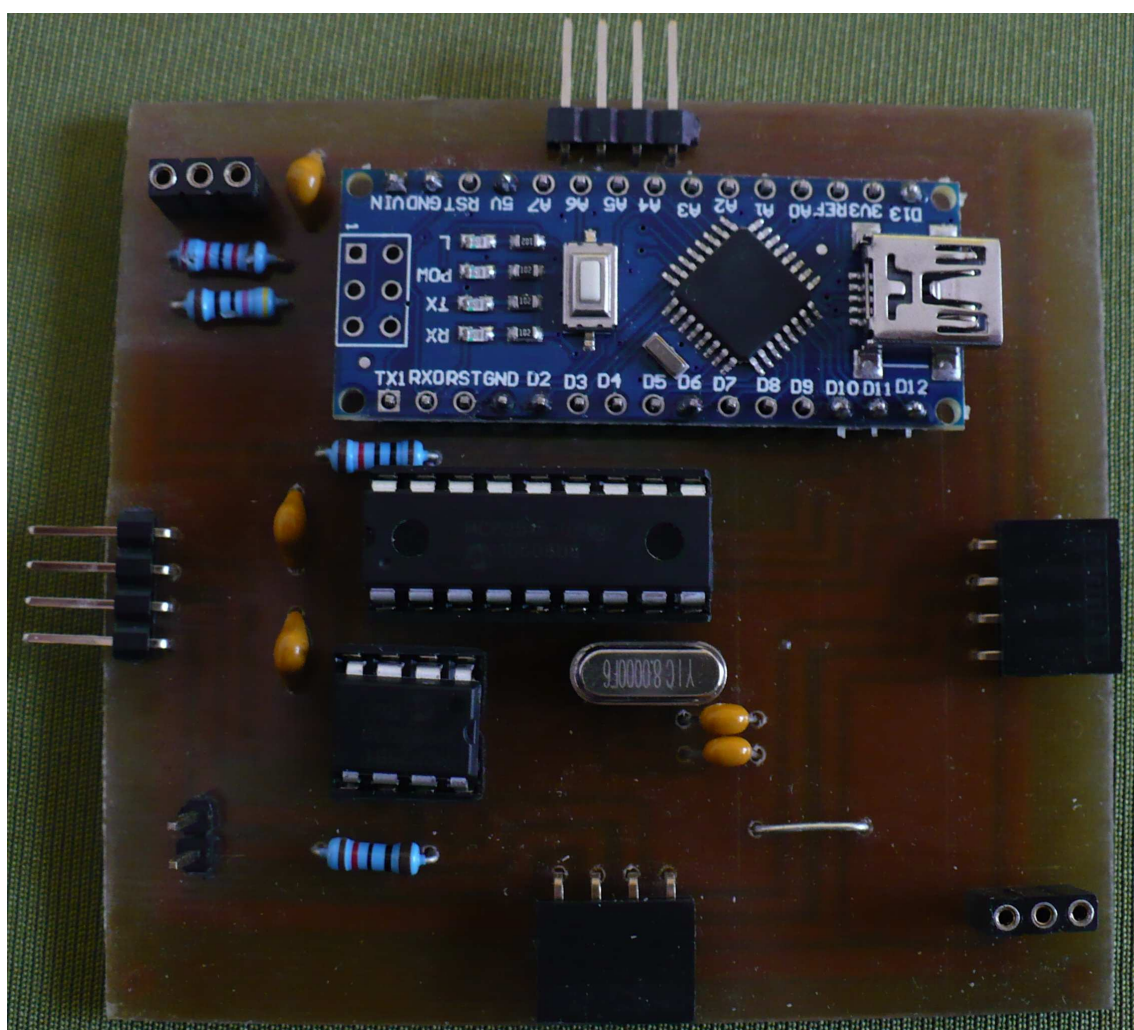
Deska plošných spojů – rozložení součástek



Obrázek C.1: DPS rozložení součástek

Příloha D

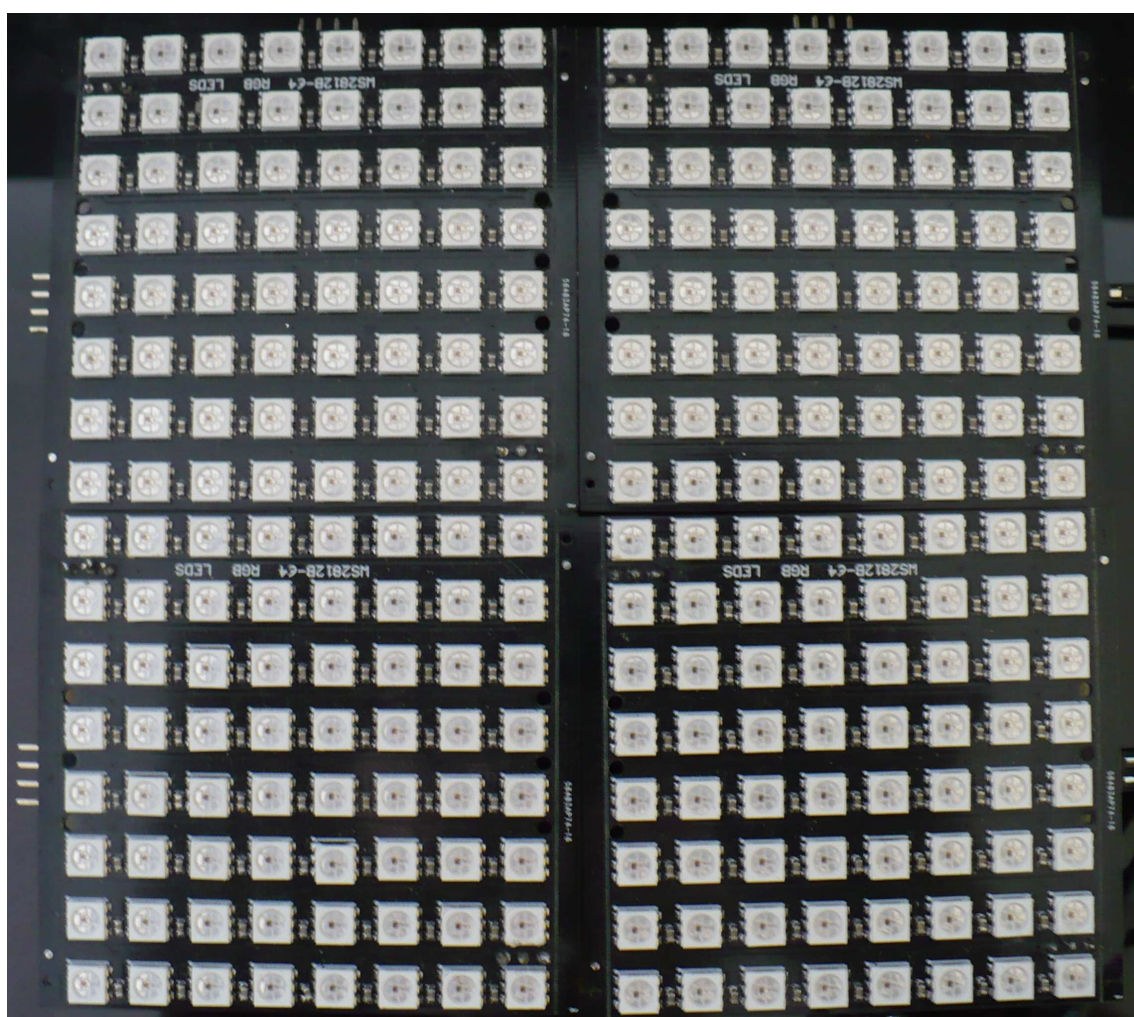
Osazený LED modul



Obrázek D.1: Osazený modul

Příloha E

Výsledný celek



Obrázek E.1: LED displej složený ze čtyř modulů