

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2021

Bc. Miroslav Petřek



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

MATICOVÝ DISPLAY S FLIP-DOT MATICÍ

FLIP-DOT MATRIX DISPLAY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Miroslav Petřek

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Bradáč, Ph.D.

BRNO 2021

Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Miroslav Petřek

ID: 174235

Ročník: 2

Akademický rok: 2020/21

NÁZEV TÉMATU:

Maticový display s flip-dot maticí

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a zrealizujte řídicí systém ke stávajícímu flip-dot zobrazovacímu panelu firmy BUSE. Systém navrhněte tak, aby byl univerzální a široce flexibilní. Systém musí umožňovat vhodnou externí komunikaci s nadřazenými systémy. Řízení navrhněte tak, aby se daly ovládat všechny pixely nezávisle.

1. Proveďte internetovou a literární rešerši informačních panelů.
2. Proveďte analýzu stávajícího stavu HW, otestujte jeho funkčnost a proveďte případné opravy. Navrhněte a zrealizujte vhodný mikrokontrolerový systém včetně všech periferních obvodů pro řízení panelu tak, aby umožňoval komunikaci s okolím přes Ethernetové a bezdrátové rozhraní.
3. Pro PC a vestavěný mikrokontrolér navrhněte a implementujte vhodné programové vybavení, které umožní zobrazování informací na připojeném panelu.
4. Zrealizujte funkční vzorek zařízení.
5. Demonstrujte a ověřte plnou funkčnost zařízení a zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6

Dle pokynů vedoucího práce.

Termín zadání: 8.2.2021

Termín odevzdání: 17.5.2021

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

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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 diplomová práce se zabývá návrhem a realizací řídicího systému, doplňujícího stávající flip-dot zobrazovací panel BS110 a umožňuje Ethernetovou a bezdrátovou komunikaci. Nejprve se krátce věnuje původnímu využití displeje, a uvádí přehled typů zobrazovacích jednotek. Poté už následuje analýza, testování a oprava panelu BS110 na niž navazuje popis navrženého konceptu a tvorba obvodového řešení řídicích počítačů a jejich obvodů. Pro ně je v práci dále popsán návrh a realizace programového vybavení sestávajícího z webových stránek, databáze, skriptu pro řízení MCU a programu pro řízení obvodů displeje. Takto navržený a sestavený systém je nakonec v práci demonstrován a zhodnocen.

Klíčová slova

informační panel, BUSE BS110, Raspberry Pi, PHP, MySQL, Atmega32, Apache, Python

Abstract

This thesis deals with the design and implementation of a control system that complements the existing BS110 flip-dot display panel and enables Ethernet and wireless communication. It first briefly discusses the original use of the display and gives an overview of the display types. This is followed by an analysis, testing and repair of the BS110 panel, followed by a description of the proposed concept and the development of the control computers and their circuitry. For these, the thesis further describes the design and implementation of the software consisting of a website, database, MCU control script and display circuit control program. Finally, the system so designed and built is demonstrated and evaluated.

Keywords

Information panel, BUSE BS110, Raspberry Pi, PHP, MySQL, Atmega32, Apache, Python

Bibliografická citace

PETŘEK, M. *Maticový display s flip-dot maticí*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/134828>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky.
Vedoucí práce Zdeněk Bradáč.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	<i>Miroslav Petřek</i>
VUT ID studenta:	<i>174235</i>
Typ práce:	<i>Diplomová práce</i>
Akademický rok:	<i>2020/21</i>
Téma závěrečné práce:	<i>Maticový display s flip-dot maticí</i>

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné 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 uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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.

V Brně dne: 17. května 2021

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Zdeňku Bradáčovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: 17. května 2021

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. TEORETICKÝ ROZBOR.....	12
1.1 PALUBNÍ INFORMAČNÍ SYSTÉMY V HROMADNÉ DOPRAVĚ	12
1.1.1 <i>Sběrnice užívané v informačních systémech</i>	<i>13</i>
1.2 INFORMAČNÍ PANELE	13
1.3 ZOBRAZOVACÍ JEDNOTKY – DISPLEJE.....	14
1.3.1 <i>Segmentové</i>	<i>14</i>
1.3.2 <i>Maticové.....</i>	<i>15</i>
1.3.3 <i>Grafické</i>	<i>16</i>
1.3.4 <i>Elektromechanické.....</i>	<i>16</i>
1.3.5 <i>Elektronické</i>	<i>19</i>
2. ANALÝZA, OVĚŘENÍ A OPRAVA BS110	22
2.1 ZOBRAZOVACÍ MODULY	22
2.1.1 <i>Analýza funkčnosti</i>	<i>22</i>
2.1.2 <i>Ověření funkčnosti</i>	<i>28</i>
2.1.3 <i>Analýza a oprava zjištěných závad</i>	<i>28</i>
2.2 NAPÁJECÍ ZDROJ.....	30
3. NÁVRH ZAŘÍZENÍ	32
3.1 ELEKTRONIKA ŘÍZENÍ ŘÁDKŮ.....	33
3.2 ŘÍDÍCÍ MINIPOČÍTAČ	36
3.3 POMOCNÝ MCU PŘEVODNÍK.....	37
3.3.1 <i>Převod napěťových úrovní</i>	<i>38</i>
3.3.2 <i>Rozšiřující konektor</i>	<i>39</i>
3.4 NAPÁJENÍ, OCHRANA PROTI NADPROUDU A PŘEPĚTÍ 5V VĚTVE.....	40
3.5 SESTAVENÍ NÁVRHU A VÝROBA HW	40
4. NÁVRH SW VYBAVENÍ.....	42
4.1 ATMEGA32	44
4.1.1 <i>Použité nástroje při tvorbě kódu</i>	<i>44</i>
4.1.2 <i>Tvorba kódu a jeho části.....</i>	<i>44</i>
4.1.3 <i>Časovač Timer2</i>	<i>51</i>
4.1.4 <i>Struktura dat uložených zobrazovaných bodů proměnné pole_pixelu</i>	<i>52</i>
4.1.5 <i>Sériová komunikace</i>	<i>54</i>
4.2 RASPBERRY PI 3 B+	54
4.2.1 <i>Použité nástroje při tvorbě kódu</i>	<i>54</i>
4.2.2 <i>Příprava RPi pomocí dostupného SW vybavení.....</i>	<i>54</i>
4.2.3 <i>Struktura internetových stránek</i>	<i>55</i>
4.2.4 <i>Popis částí internetových stránek.....</i>	<i>55</i>
4.2.5 <i>Databáze</i>	<i>61</i>
4.2.6 <i>Vytváření řídicího skriptu v jazyce Python</i>	<i>62</i>

5. DEMONSTRACE, OVĚŘENÍ A ZHODNOCENÍ	65
5.1 PŘÍPRAVA PARAMETRŮ A TESTOVACÍCH DAT	65
5.2 DEMONSTRACE A TESTOVÁNÍ FUNKCÍ	66
5.2.1 <i>Internetové stránky</i>	66
5.3 VÝSLEDKY ZOBRAZENÍ NA DISPLEJI	67
5.4 ZHODNOCENÍ DOSAŽENÝCH VÝSLEDKŮ	67
6. ZÁVĚR.....	69
SEZNAM SYMBOLŮ A ZKRATEK	74
SEZNAM PŘÍLOH.....	75

SEZNAM OBRÁZKŮ

Obr. 1.1 Možné uspořádání palubního systému IV. generace s veřejným internetem [1]	12
Obr. 1.2 Způsob zobrazování na alfanumerických displejích [2]	14
Obr. 1.3 Ukázka možných zobrazovaných symbolů [4]	15
Obr. 1.4 Míchání barev RGB	16
Obr. 1.5 Vzhled a princip funkce překlápěcího displeje [6]	17
Obr. 1.6 Vzhled a princip funkce „vane“ displeje [6]	18
Obr. 1.7 Vzhled a princip funkce terčíkového displeje [6]	18
Obr. 1.8 Příklad využití digitronů jako zobrazovače hodin [12].....	19
Obr. 1.9 Ukázka vnějšího vozidlového čelního LED panelu [15]	20
Obr. 1.10 Ukázka LCD informační tabule [16]	21
Obr. 2.1 Ukázka jednoho pásků („stripsu“) [08].....	22
Obr. 2.2 Schéma principu zapojení matice cívek displeje	23
Obr. 2.3 Popis kontaktů propojovacího 50pinového konektoru modulů BS110.....	24
Obr. 2.4 Principiální schéma funkce výběru sloupce modulu displeje	25
Obr. 2.5 Ukázka vadný pixelů – černá strana	27
Obr. 2.6 Ukázka vadný pixelů – žlutá strana	27
Obr. 2.7 parametry diody BAV99 [27]	28
Obr. 3.1 Blokové schéma návrhu zařízení	32
Obr. 3.2 Zapojení a princip funkce H-můstku [17].....	33
Obr. 3.3 Blokový diagram vnitřního zapojení obvodu L293D	34
Obr. 3.4 Obvod pro výběr 1 z 19 řádků	35
Obr. 3.5 Označení pinů Raspberry Pi konektoru GPIO [19]	37
Obr. 3.6 Hodnoty logických napěťových úrovní [20].....	38
Obr. 3.7 Příklad zapojení napěťového převodníku	39
Obr. 3.8 Schéma zapojení napájení.....	40
Obr. 3.9 Výsledný vzhled vyrobeného HW	41
Obr. 4.1 Sestavení a způsob komunikace SW vybavení Rpi	42
Obr. 4.2 Znázornění řízení na straně převodníku.....	43
Obr. 4.3 Vývojový diagram main.c	45
Obr. 4.4 Vývojový diagram zobraz_cely_disp	46
Obr. 4.5 Vývojový diagram přichozí komunikace.....	46
Obr. 4.6 Vývojový diagram zobraz_dva_radky	47
Obr. 4.7 Vývojový diagram HW_vytvor_disp.....	49
Obr. 4.8 Vývojový diagram HW_zapis	50
Obr. 4.9 Způsob uložení fontu v MCU	51
Obr. 4.10 Výpočet užívaný při binárním zápisu do proměnné	53
Obr. 4.11 Způsob uložení informace o obrazu displeje	53
Obr. 4.12 Ukázka stránky login.php	57
Obr. 4.13 Ukázka stránky main.php	58
Obr. 4.14 Ukázka stránky prime.php	59
Obr. 4.15 Ukázka stránky projekty.php	60
Obr. 4.16 Znázornění tabulek databázi a jejich vazby	62
Obr. 5.1 Ukázka obrazovek projekty „demo_bmp_PM“	65
Obr. 5.3 Ukázka zobrazení textových dat	68
Obr. 5.2 Ukázka zobrazení obrazových dat	68

SEZNAM TABULEK

Tab. 2.1 Tabulka naměřených hodnot úbytků napětí diod BAV99	29
Tab. 3.1 Porovnání parametrů při výběru H-můstku	34
Tab. 3.2 Hodnoty výstupu obvodu L293D v závislosti na signálu EN	35
Tab. 3.3 Hodnota řídicího signálu pro výběr řádku	36
Tab. 3.4 Tabulka základních parametrů použitého minipočítače [18]	36
Tab. 3.5 Porovnání napěťových úrovní při přenosu z Raspberry Pi do ATmega32	38
Tab. 3.6 Porovnání napěťových úrovní při přenosu z ATmega32 do Raspberry Pi	39
Tab. 5.1 Tabulka přístupových údajů pro demonstraci funkce	65

ÚVOD

Tato diplomová práce se zabývá využitím zobrazovací jednotky jednoho ze starších informačních panelů nesoucího označení „BS 110.19.140.1“ od firmy BUSE s.r.o. Ten v minulosti patřil k součástem informačního systému prostředků hromadné dopravy, jako jsou např. autobusy, trolejbusy či tramvaje. Sám o sobě však již není schopen vlastního fungování, a proto vznikl požadavek na jeho doplnění o části nezbytné pro jeho funkční využití mimo jeho původní informační systém.

Mezi hlavní cíle této práce patří zanalyzovat, otestovat a opravit již existující elektroniku zobrazovací jednotky. Pro ni následně provést návrh a realizaci obvodového a programového vybavení, které umožňuje vhodnou vnější komunikaci pomocí Ethernetového a bezdrátového rozhraní a řízení všech pixelů nezávisle na sobě. Nakonec takto navržený a sestavený funkční vzorek zařízení demonstrovat, ověřit jeho plnou funkčnost a zhodnotit dosažené výsledky.

Na základě stanovených cílů byl pro zobrazovací jednotku nejprve proveden krátký teoretický rozbor následovaný rozbohem součástí daného panelu a jejich zapojení, na jehož základě je možné pochopit princip funkce řízení matice zobrazovacích bodů. Následně byl navržen koncept a obvodové řešení chybějících částí zobrazovací jednotky. Obvodové řešení se sestává z obvodů pro řízení řádků, řídicího mikrokontroléru ATmega32 a jemu nadřazeného mini počítače Raspberry Pi. Pro obě programovatelné jednotky bylo navrženo a sestaveno programové vybavení. Pro mikrokontrolér se skládá z částí pro ovládání displeje a komunikaci s nadřazeným mini počítačem. Pro nadřazený mini počítač se skládá z webu, databáze a obslužných programů. Následně byl systém jako celek demonstrován a zhodnocen.

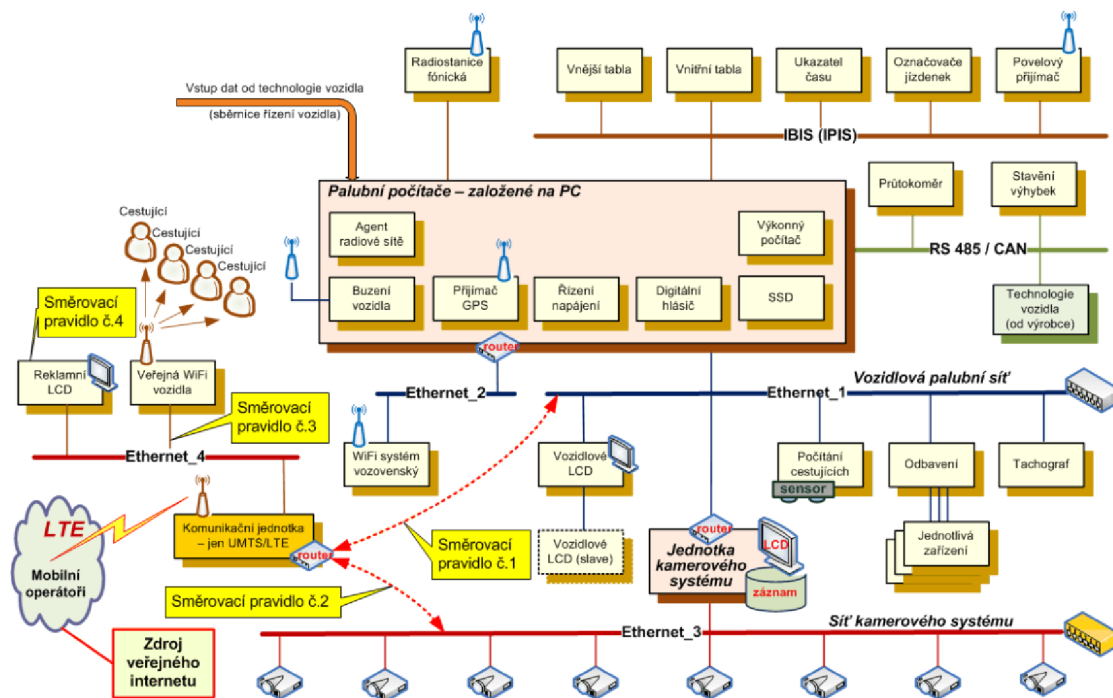
Text práce je rozdělen do pěti tematických celků, z nichž 1. kapitola s názvem Teoretický rozbor se obecně věnuje informačním panelům a jejich zobrazovacím jednotkám. Kapitola 2 se zabývá analýzou, a ověřením funkce elektroniky zobrazovací jednotky panelu BUSE 110 a následnou analýzou a opravou zjištěných závad. V kapitole 3 je popsán postup při návrhu a realizaci elektronických obvodů na niž navazuje kapitola 4 popisující návrh a realizaci programového vybavení. Funkce výsledného zařízení demonstruje a hodnotí kapitola 5

1. TEORETICKÝ ROZBOR

V dnešní době nás obklopuje velké množství informací, a to nejen doma či v práci ale i na veřejnosti. Některé užitečné informace nám často mohou být poskytovány prostřednictvím informačních panelů a jiných obrazovek různých typů. Ty jsou součástí často složitých informačních systémů usnadňujících přehled o dané situaci. Nejčastěji se vyskytují ve veřejné dopravě, kde lze najít informační tabule například o příjezdech a odjezdech spojů. Lze je ale najít i na mnoha dalších místech, jako jsou třeba parkoviště, kde ukazují aktuální obsazenost, či na úřadech, kde informují o pořadí obsluhy u jednotlivých přepážek.

1.1 Palubní informační systémy v hromadné dopravě

Informační panel přidělený pro tuto práci patřil k součástem palubního informačního systému vozidel hromadné dopravy. Ty mají v české republice počátky již v druhé polovině 90 let minulého století. Slouží pro jasné, srozumitelné a rychlé informování cestujících o aktuálním stavu, místě a čase. Kromě informačních panelů byly paluby vozidel hromadné dopravy v průběhu let doplňovány i o další užitečné funkce jako



Obr. 1.1 Možné uspořádání palubního systému IV. generace s veřejným internetem [1]

například vnější rádiové komunikace umožňující automaticky vyměňovat data uložená v systému, či systémy odbavení cestujících. Moderní palubní systémy dnes již nejen, že umožňují kontrolu důležitých částí vozu a poskytování dat o jejich stavu, snímání

interiéru kamerovými systémy, či poskytování veřejného připojení k internetu pomocí Wifi, jsou dokonce součástí komplexního informačního systému, v němž komunikují po celou dobu s nadřazenými systémy dopravního podniku pomocí bezdrátových komunikací jako jsou GSM a LTE. Díky tomu dnes umožňují například prioritizaci čili upřednostňování určitých vozidel v provozu křižovatek. Na obr. 2.1 je možné vidět ukázkou komplexnosti moderních palubních systémů. Realizace moderních palubních systémů v České republice provádí několik různých firem jako jsou např. BUSE s.r.o., či APEX spol. s r.o. [1]

1.1.1 Sběrnice užívané v informačních systémech

V rámci palubních systémů se již od počátku využívá komunikace skrze vodičové sběrnice. Toto jsou tři vybrané příklady:

- Sběrnice *IBIS*:

Vzhledem ke svému stáří využíval panel BS110 ještě první a dodnes užívanou sběrnici IBIS. Ta je již od počátku využívaným standardem pro evropskou veřejnou dopravu. Díky tomu umožňuje propojení většiny standardních periférií. Jeho komunikační rychlost je však omezena na pouze 1200 bitů za sekundu, a proto ji využívaly jen nenáročná zařízení jako jsou právě informační tabule neboli tabla. V české republice je jeho česká verze označována jako IPIS.

- RS 485:

Je první tzv. rychlá sběrnice. Díky její komunikační rychlosti 115,2 kbit/s bylo možné do vozů přidat nové funkce jako komunikace s centrálním dispečinkem a aktualizace dat ve vozovkách. Toho bylo docíleno skrze připojení vnější radiové komunikace.

- Sběrnice *Ethernet*:

V rámci zadání této práce je potřeba využít připojení pomocí sběrnice Ethernet. V současných palubních systémech je tato sběrnice taktéž využívána, a to zejména pro velmi rychlé aplikace a přenosy velkoobjemových dat. Jmenovitě jsou to zejména zabudované vozidlové LCD, tachografy, kamerové systémy, připojení veřejné sítě, popř. i systém odbavení.

1.2 Informační panely

Základní funkcí informačního panelu je zobrazovat informace, a to nejčastěji v textové či číselné podobě. Zobrazení může probíhat buď staticky, jako je tomu například u starších tabulí příjezdů a odjezdů spojů hromadné dopravy, nebo dynamicky, například ve formě rolujícího textu či animace, jako je tomu u moderních LED displejů.

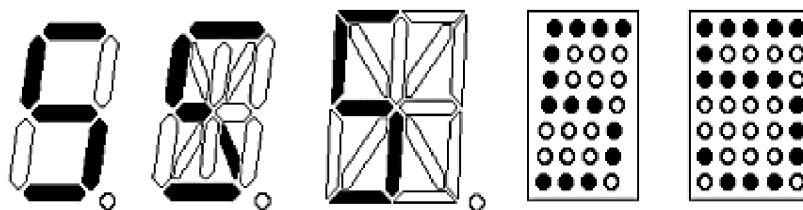
Hlavní částí informačních panelů je tedy zobrazovací jednotka – displej. Těch dnes existuje nepřeberné množství typů, lišící se užitím, velikostí, rozlišením, cenou, použitými technologiemi a dalšími parametry, jak je možné vidět v následující podkapitole.

1.3 Zobrazovací jednotky – displeje

Jak již bylo zmíněno, zobrazovací jednotka (neboli displej) je hlavní součástí informačních panelů, neboť ji každý uživatel vnímá z celého panelu nejvíce. Zobrazovací jednotky lze rozdělit například podle způsobu zobrazování na segmentové, maticové a grafické [3], či podle principu funkce na elektromechanické a elektronické. Elektronické displeje lze dále dělit podle použitých technologií jako jsou například LED, LCD, VFD, CRT a mnoho dalších. Následující část podkapitoly slouží jako obecný přehled vlastností a typů zobrazovacích jednotek. Nejprve jsou zde rozděleny typy podle způsobu, jak je zobrazovaná informace interpretována (segmenty a body) a poté následuje dělení podle technologie. Nejprve jsou zde zmíněny starší tzv. mechanické displeje a poté následují již novější označované jako elektronické.

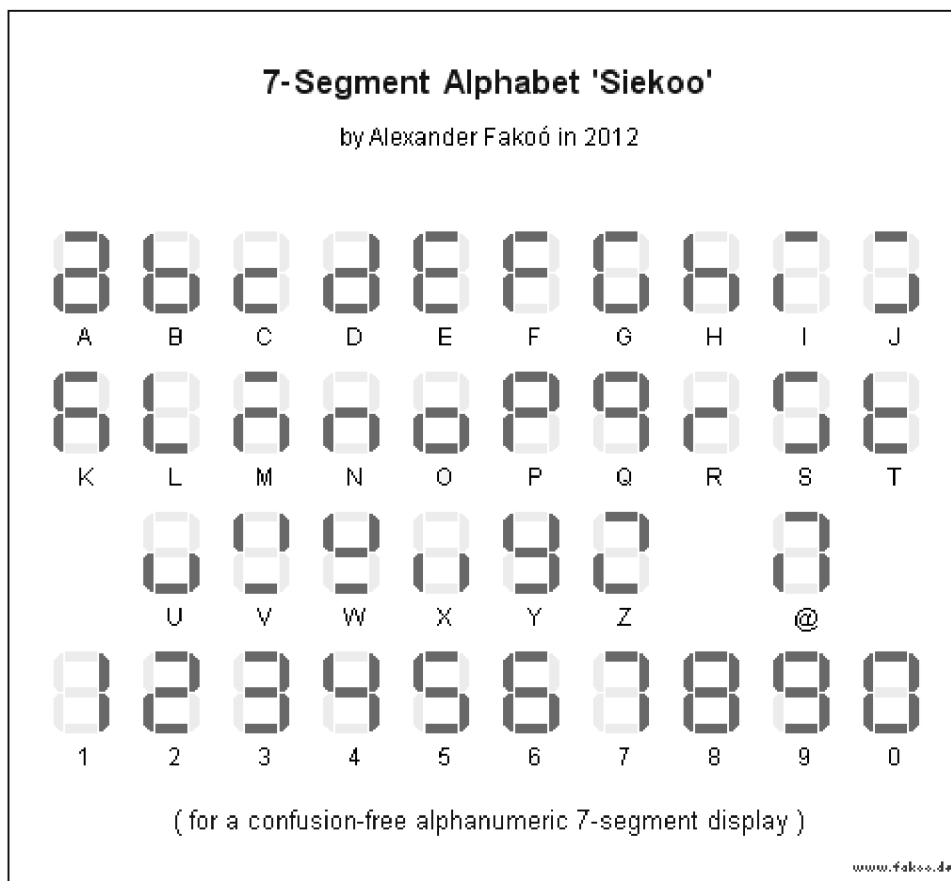
1.3.1 Segmentové

Segmentové displeje jsou typické svým nízkým počtem zobrazovacích bodů nazývaných segmenty. Z nich jsou často sestaveny jednotlivé zobrazované číslice či znaky. Když se řekne segmentový displej, nejčastěji se každému vybaví 7segmentový LED displej. Ten je sice snad nejčastěji využívaným typem, ale existuje mnohem více typů segmentových displejů, lišících se nejen počtem segmentů ale i použitou technologií. Kromě již zmíněné technologie LED, existují i LCD segmentové displeje, či třeba starší VFD displeje.



7-segment 14-segment 16-segment 4x7 matrix 5x7 matrix
Obr. 1.2 Způsob zobrazování na alfanumerických displejích [2]

Displeje se sedmi segmenty jsou určeny primárně pro zobrazování arabských číslic. Lze na nich ale zobrazovat i upravené verze abecedy. Ty sice pro některá písmena nevypadají úplně intuitivně, jako například písmeno „m“, ale i přesto jsou užívána například v levné spotřební elektronice, jako jsou satelitní přijímače nebo tzv. televizní „set top boxy“. Na obr. 2.2 je vidět jeden ze způsobů zobrazení abecedy na 7segmentovém displeji. Pro intuitivnější vzhled znaků abecedy existují 14segmentové a 16segmentové displeje. U nich již lze rozeznat nejen malá a velká písmena, ale například i závorky a další užívané znaky.



Obr. 1.3 Ukázka možných zobrazovaných symbolů [4]

1.3.2 Maticové

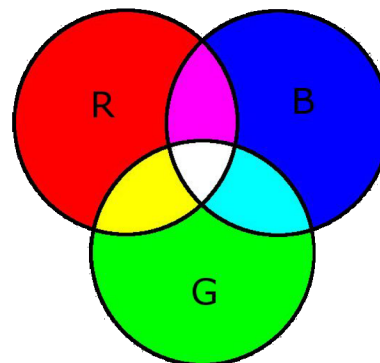
Další možností je zbavit se segmentů různého tvaru a rozmístění a přejít na maticové uspořádání. Pro příklad třeba na matice čítající 4×7 , nebo 5×7 bodů, jak je vidět na obr. 2.1. Souhrnně se obou zmíněným typům těchto displejů říká alfanumerické, protože jsou určeny pro zobrazování čísel a znaků. Pro zobrazování tvarů a obrazů už stačí jen zvýšit počet zobrazovacích bodů a scelit do jedné souvislé plochy - vzniká tak tzv. grafický displej.

1.3.3 Grafické

Grafické displeje nám již konečně umožňují, kromě zobrazování alfanumerických informací, zobrazovat i libovolné tvary, obrazce, grafy a další bodově reprezentované obrazy. Aby však zobrazované obrazce vypadaly věrohodně, musí mít tyto displeje z pravidla velký počet zobrazovacích bodů. (Statisíce až miliony) Pro zobrazovací bod se využívá anglického slova „pixel“ (zkrácení anglických slov picture element) a počet těchto pixelů se označuje jako rozlišení. Kromě něj je důležité i jejich rozmístění, tedy tzv. poměr stran.

Mezi dnes nejrozšířenější poměry patří 4:3, 16:9 a 16:10. Grafické displeje existují jako monochromatické

(často chybně označované jako černobílé), nebo barevné. U barevných displejů dochází k tzv. míchání výsledné barvy pixelu zpravidla z trojice různobarevných zobrazovacích bodů (nejčastěji červená, zelená a modrá), jejichž kombinace umožňuje zobrazení většiny barevného spektra. [4]



Obr. 1.4 Míchání barev RGB

1.3.4 Elektromechanické

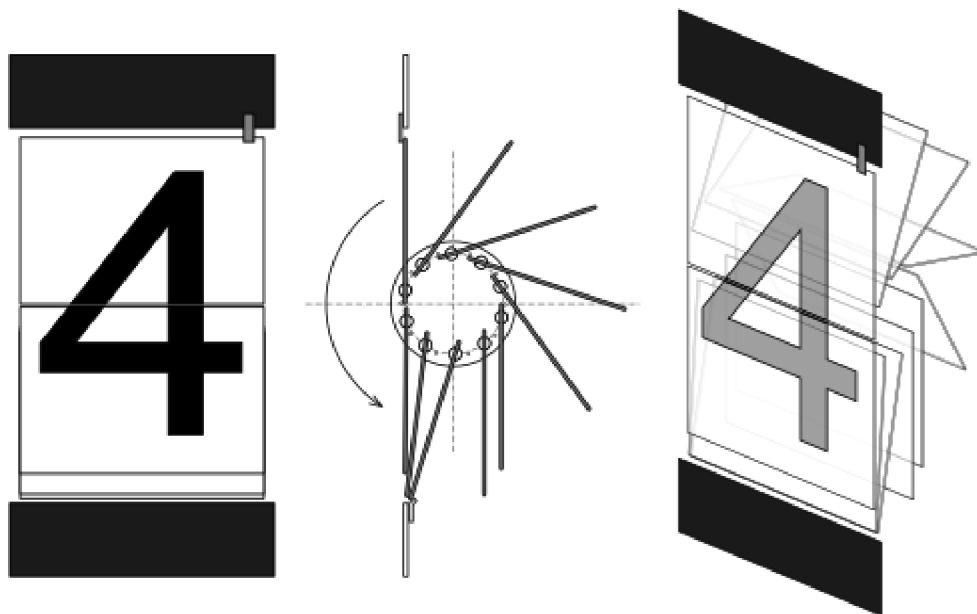
Z historického vývoje mezi první zobrazovací jednotky informačních panelů patří elektromechanické displeje. Ty jsou převážně napájeny elektrickým proudem, který je skrze elektromagnetické síly převáděn na mechanický pohyb. Ten následně způsobuje změny zobrazovaných informací. Díky tomu není pro zachování změn zobrazené informace zapotřebí žádná energie.

Vyznačují se zpravidla dobrou čitelností za bílého dne, vysokým kontrastem daným barvou zobrazovacích ploch, velkými pozorovacími úhly a již zmíněným nulovým klidovým příkonem. Poslední ze jmenovaných vlastností má za důsledek výše zmiňovaný princip funkce. Samotné zobrazení tedy probíhá pasivně (respektive s využitím odrazu okolního světla), a proto lze tyto displeje považovat podle principu zobrazování za tzv. pasivní. Z této výhody se však za zhoršených světelných podmínek stává nevýhoda a vzniká potřeba dalšího zdroje osvětlení. Protože jsou však kvůli mechanickému principu poměrně složité na výrobu, a obsahují pohyblivé součásti, které mohou časem vyžadovat údržbu, jsou dnes často nahrazovány modernějšími elektronickými displeji.

- **Překlápěcí**

Za nejjednodušší typ elektromechanické zobrazovací jednotky lze považovat dělené překlápěcí displeje anglicky nazývané „split-flap“. Jedná se o soubor za sebou řazených pevných ploch, na kterých se z obou stran nachází na půl dělené číslice, písmena či jiné symboly a obrázky.

Změna zobrazované informace probíhá skrze otáčení středové hřídele, na které jsou listy připevněny. Pozice každé zobrazované strany je dána úhlem otočení hřídele. Tento princip změny zobrazované informace způsobuje onen typický zvukový projev. Nejlépe je celý tento princip vidět na obr. 2.4.

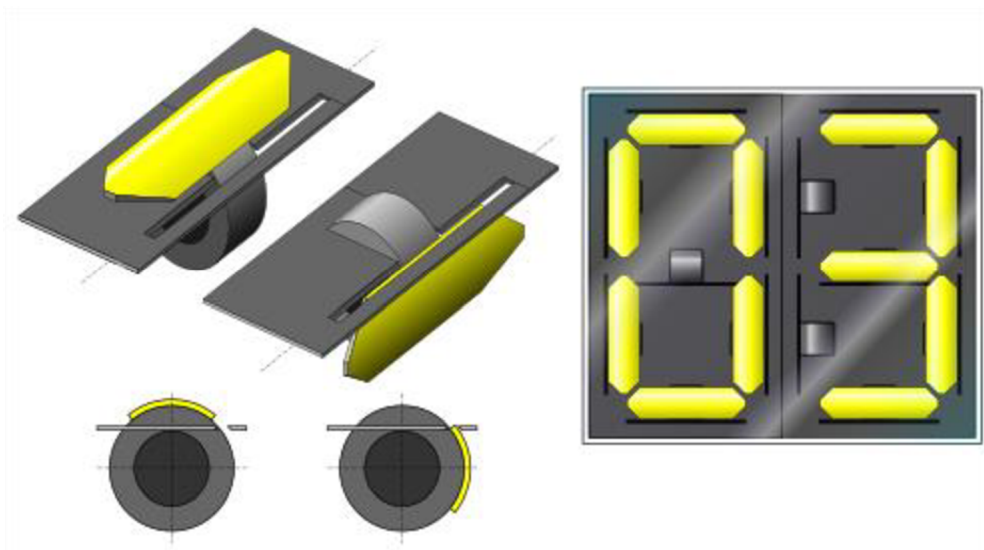


Obr. 1.5 Vzhled a princip funkce překlápěcího displeje [6]

Tyto displeje se nejčastěji používaly jako informační tabule ve veřejné dopravě (na letištích, vlakových nádražích apod.) nebo například v budících pro zobrazování hodin. (tzv. „flip clock“) U nás nejznámějším výrobcem těchto jednotek byl minulosti výrobní podnik Pragotron (dnes Elektročas – Pragotron). [7] Mezi hlavní nevýhody spojené s mechanickou konstrukcí patří spolehlivost a hluk spojený s překlápěním.

- **Segmentové („Vane“) displeje**

Jde o jeden z typů sedmi segmentových elektromechanických displejů. Název těchto panelů je dán anglickým slovem „vane“, které v překladu zní vanička, a to pro tvar jeho prohnutých segmentů z kterých se displej skládá. Tento tvar zaručuje minimální otvor v panelu displeje potřebný pro ukrytí nezobrazených segmentů, které způsobuje jejich rotace po kružnici. Toho si lze všimnout na obr. 2.5. Samotná změna informace tedy probíhá opět mechanickým pohybem způsobeným magnetickým polem cívky. V některých případech docházelo místo rotace spíše k překlápění, a tak se tyto typy označují slovem „flip“. Kromě takto řízených displejů se dříve využívaly i informační tabule, kde tuto změnu prováděla obsluha odklápěním jednotlivých segmentů. Používaly se dříve například výsledkové tabule, či pro zobrazování cen pohonných hmot u čerpacích



Obr. 1.6 Vzhled a princip funkce „vane“ displeje [6]

stanic.[8]

- **Terčikové displeje**

Na rozdíl od předešlých typů mají terčikové displeje možnost zobrazení libovolných znaků nebo textu, a to skrze matici složenou z menšího počtu bodů (řádově do jednotek tisíců). Tyto body jsou česky označovány jako terčiky a mají 2 strany – jednu světlou a jednu tmavou. Obecně jsou známé jako „flip-dot“ či „flip-disc“ displeje. (Z angličtiny otočný bod, či otočný disk)

K samotné změně zobrazované informace dochází prostřednictvím překlápění jednotlivých terčiků, z nichž každý obsahuje malý magnet. Samotného otáčení je docíleno pomocí krátkého proudového impulsu tekoucího skrze cívku, která se nachází se v magnetickém obvodu každého terčiku. Podle směru toku proudu dojde buď k zápisu daného bodu, nebo naopak. Zápisem je myšleno zobrazení světlé strany terčiku a naopak při tzv. vymazání dojde k zobrazení matně černé strany terčiku. Lze to ale samozřejmě i obrátit a zapisovat tmavé body do světlého pozadí.



Obr. 1.7 Vzhled a princip funkce terčikového displeje [6]

U nás je originálním výrobcem patentovaných elektromagnetických zobrazovacích prvků s otočnými terči firma BUSE s.r.o., která navíc nabízí i panely s vlastní patentovanou technologií označovanou „DOT-LED“, kdy má každý bod svou vlastní LED pro zajištění optimálních podmínek i v noci. [10]

Pro zajímavost, existuje společnost BREAKFAST, která tuto technologii a její hardware dovedla téměř k dokonalosti s ohledem na rychlost zobrazování změn. Prostřednictvím těchto displejů vytváří originální umělecké instalace. [11]

1.3.5 Elektronické

Na rozdíl od elektromechanických, kdy zobrazování informací probíhá díky odrazu okolního světla, elektronické displeje zpravidla světlo samy vyzařují, a to ať už přímo, nebo nepřímo pomocí podsvícení. Díky tomu jsou považovány podle principu zobrazování za tzv. aktivní. K vyzařování dohází buď přímo, kdy se rozsvěcují jednotlivé segmenty či body (LED, VFD, OLED atd.), nebo skrze podsvícení potřebné pro správnou funkci displeje (LCD, popř. novější typy označované „LED“ lišící se pouze technologií podsvícení využívající LED namísto CCFL). Za výjimku lze považovat pasivní LCD displeje, využívané u nízko odběrových zařízení, často napájených bateriemi, jako například kalkulačky, hodinky, meteostanice apod.

- **Znaková výbojka (digitron)**

Digitrony stojí za to zmínit, i když pro svou velikost je takřka nelze použít jako informační panel, protože jsou to první elektronické zobrazovače. Jde vlastně o plynem plněnou výbojku se studenou katodou. Ta je v provedení buď jako číslicová výbojka pro zobrazování číslic, popř. jako znaková výbojka pro zobrazování znaků dalších symbolů. U nás jsou známy jako digitrony, kdežto v anglicky mluvících zemích jsou označovány názvem „Nixie tubes“.

Vzhledem se sice výrazně podobá elektronkám, ale co do principu funkce jde o



Obr. 1.8 Příklad využití digitronů jako zobrazovače hodin [12]

doutnavku se zpravidla několika oddělenými katodami, které tvoří jednotlivé zobrazované znaky. Do těchto katod je proud přiváděn ze společné anody tvořené tenkou

mřížkou. Díky zmíněnému proudu se plyn (neon) v okolí katody rozzáří doutnavým světlem. [13]

Během 50. až 70. let byly využívány jako zobrazovače v raných číslicových přístrojích, tedy například v hodinách, měřicích přístrojích apod. Brzy je však nahradily modernější LED, VFD či LCD displeje. I přesto se dnes stále v omezeném množství vyrábí pro svůj jedinečný retro vzhled. [14]

- **LED displej**

LED displeje se skládají z velkého množství diod označovaných jako LED. Toto označení pochází z anglického spojení slov „Light-Emitting Diode“, které se do češtiny překládá jako elektroluminiscenční dioda, nebo také svítivá dioda. Jde o typ elektronické diody, která je schopna emitovat světelné záření. Samotný princip spočívá v průchodu



Obr. 1.9 Ukázka vnějšího vozidlového čelního LED panelu [15]

stejnoseměrného elektrického proudu v propustném směru skrze PN přechod každé svítivé diody.

LED dnes dosahují vysoké účinnosti, dobré mechanické odolnosti a díky zvládnuté technologii výroby i nízké ceny. Díky těmto výhodám jsou hojně využívány v širokém spektru aplikačních oblastí, a to nejen jako displeje, ale například i jako zdroje světla pro osvětlování či podsvícení LCD displejů atd. Displeje schopné jemného grafického zobrazení pracující na principu svítících diod se označují OLED.

Samotné LED panely jsou tedy cenově dostupné a dosahují velmi nízké spotřeby. I díky antireflexním úpravám krycích skel zabraňujících odleskům dosahují velmi dobré čitelnosti i při přímém slunečním svitu. I přesto, že je dnes ještě snaha modernizovat terčíkové displeje (viz. kapitola 2.2.4.3 Terčíkové displeje) jsou dnes nově montovány spíše zobrazovače založené čistě na bázi LED. Jelikož neobsahují pohyblivé části, nevyžadují takřka žádnou údržbu při dosahování vysoké životnosti, a tedy dobré návratnosti pořizovacích nákladů.

- **LCD**

Zkratka LCD pochází ze spojení anglických slov „Liquid crystal display“, které se do češtiny volně překládá jako displej z tekutých krystalů. Jak z názvu vyplývá jde o displej složený z buněk naplněných tekutými krystaly. Ty jsou obklopeny z obou stran polarizačními filtry, jejichž polarizace je na sebe kolmá, a tedy za normálních podmínek brání průchodu světla. Samotné krystaly jsou řízeny dvěma průhlednými elektrodami umožňujícími jejich otáčení. To způsobuje kolmou změnu polarizace světla, což má za

ODJEZDY VLAKŮ
DEPARTURES OF TRAIN

CAS Time **15:14**

Druh vlaku Type of train	Číslo vlaku Number of train	Dopravce Carrier	Číslo stanice Station number	Směr jízdy Way	Plánovaný odjezd Scheduled departure	Míst. Platform	Kolej Track	Zpoždění (min.) Delay (min.)
SP	1957	ČD	LETOHRAD	Častořov.	15:31			
Os	5635	ČD	CHOCEŇ	Borohrádek	15:33			
SP	1866	ČD	HR.KRÁLOVÉ HL.N.	Třebech.p0	15:37			
Os	5134	ČD	NÁCHOD	N.Město nM	15:40			
Os	5171	ČD	DOUDLEBY N.ORL.	Častořov.	16:07			
Os	5206	ČD	CHLUMEC N.CIDL.	Třebech.p0	16:08			
SP	1867	ČD	CHOCEŇ	Borohrádek	16:31			
SP	1956	ČD	HR.KRÁLOVÉ HL.N.	Třebech.p0	16:31			

Obr. 1.10 Ukázka LCD informační tabule [16]

důsledek umožnění jeho průchodu druhým z pořadí filtrů. Pro svou funkci tedy navíc potřebuje světlo, které může pocházet buď z okolí (využití odrazu), nebo z vlastního podsvícení, nebo jejich kombinace.

Informační tabule využívající LCD panel jsou například univerzální grafické displeje často zobrazující např. reklamní sdělení, či jednoúčelové tabule jako dnes můžeme vidět třeba na vlakových stanicích. Jsou to například alfanumerické LCD informační panely řady GL od výrobce ELEKON s.r.o., který je vidět a obr. 2.9. Mezi jejich výhody patří nízká spotřeba a výborný kontrast.

• Další typy displejů

Kromě výše zmíněných existují samozřejmě i další typy displejů. Jejich rozšíření na poli informačních panelů je dnes žádné či mizivé, a proto jsou zde jen vyjmenovány.

- Vakuový fluorescenční displej (VFD)
- CRT displej (Cathode-ray tube)
- PDP (Plasma display panel)
- ELD (Electroluminescent Display)
- Elektronický papír (e-paper / e-ink)

2. ANALÝZA, OVĚŘENÍ A OPRAVA BS110

2.1 Zobrazovací moduly

Jako zobrazovací jednotka byl pro tuto práci vybrán informační panel od výrobce BUSE s.r.o., který původně sloužil jako tzv. boční informační panel umístěný v interiérech prostředků hromadné dopravy, převážně v autobusech. Podle výrobního štítku se jedná o typ BS 110.19.140.1 vyrobený v roce 1999. V rámci této práce bude zkráceně označován jako **BS110**. Z typového označení lze vyčíst, že jde o panel složený ze zobrazovacích bodů (terčků) o velikosti 10 mm, které jsou uspořádány v matici čítající 19 řádků a 140 sloupců. Celkem jich je tedy 2660. Po nahlédnutí do kovového rámu panelu je jasné vidět, že se ve skutečnosti skládá z 5 identických modulů osazených na DPS a připojených horizontálně vedle sebe. Zobrazovací strana každého z modulů se navíc skládá z matice terčků vyráběných v provedení po 7 kusech vedle sebe v jednom pouzdře tzv. „štripsu“. Tak se podle výrobce říká pásku, což je prý převzato z anglického



Obr. 2.1 Ukázka jednoho pásků („štripsu“) [08]

slova „strip“. [08] Terčíky jsou opatřeny z tmavé strany matně černou barvou a ze světlé strany výraznou žlutou. Na jedné straně plechového rámu se nachází clona pro možnost použití přisvícení displeje. Základní princip funkce a další charakteristické rysy terčíkových panelů jsou uvedeny v předchozí podkapitole.

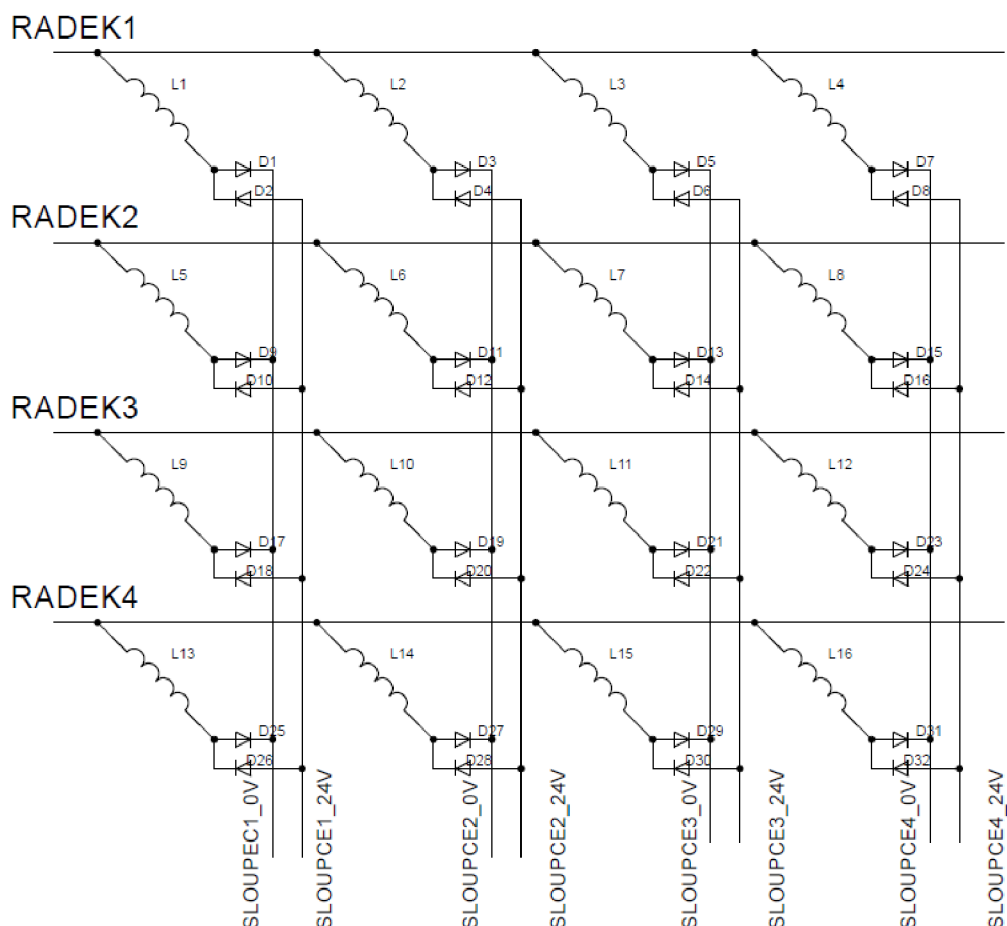
Všechny moduly jsou propojeny mezi sebou paralelně skrze 50 žilový vodič starající se o napájení a řízení panelu. Skrze něj je k modulům v původní konstrukci připojena i vlastní řídicí elektronika starající se o zobrazování informací cestujícím v prostředcích hromadné dopravy. Tato řídicí elektronika je však určena pro fungování v systémech užívaných ve vozech pro zobrazování informací o linkách a jejich zastávkách a nebyla již součástí tohoto panelu.

2.1.1 Analýza funkčnosti

Každý z pětice zobrazovacích modulů se skládá z matice bodů osazených na přední straně DPS. Celkem obsahuje 19 řádků a 28 sloupců, tedy 532 cívek, kdy každá má 2 vývody. Matice bodů je propojena tak, aby bylo možné ovládat každý jednotlivý bod za pomoci co nejmenšího počtu vodičů. Toho je docíleno tak, že první vývod všech cívek je spojen do společných řádků a druhý vývod všech cívek je spojen do společných sloupců, jak je

naznačeno na obr. 2.11. Na obrázku je možno si dále všimnout dvojice usměrňovacích diod z nichž pro každý společný sloupec vychází hned 2 vodiče.

Na zadní straně DPS se v levém a pravém dolním rohu nachází dva 50 pinové konektory pro snadné propojení s okolními zobrazovacími moduly. Oba konektory mají většinu kontaktů propojenu paralelně s výjimkou pinů 40, 41, 42. Pro lepší orientaci lze piny rozdělit do 5 hlavních skupin.

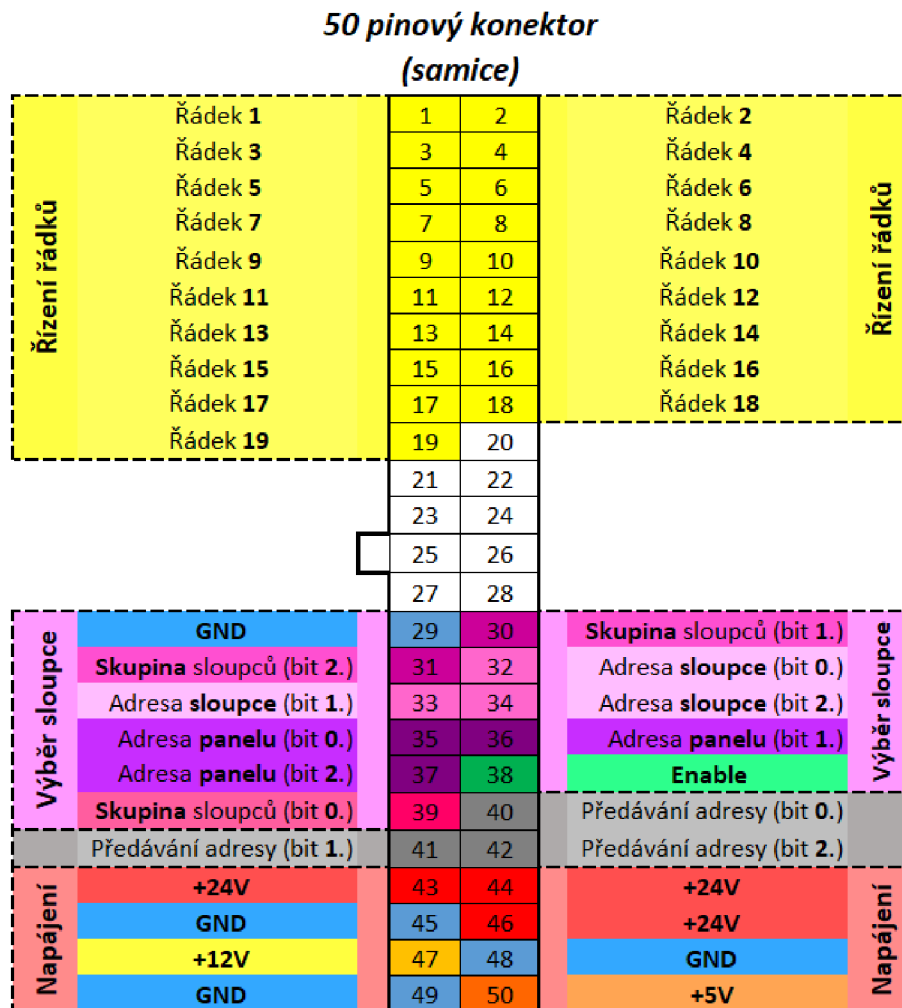


Obr. 2.2 Schéma principu zapojení matice cívek displeje

- 1) V první skupině je jen jednoduše vyvedeno všech **19 vodičů** spojujících matici do **řádků**. Jde tedy o tzv. výkonové vedení.
- 2) Ve druhé skupině se nachází **9 vodičů**, které slouží pro **oddělení číslicové části** od zmíněné výkonové.
- 3) Ve třetí skupině je **11 pinů** pro ovládání úrovně napětí v jednotlivých **sloupcích**.
- 4) Ve čtvrté se nachází **3 vodiče** přenášející **pořadovou adresu** zobrazovacího modulu danou posloupností zapojení panelů. Slovem „pořadovou“ je myšleno, že adresa odpovídá pozici modulu počítané od levé strany zobrazovacího panelu. Jde

o 3bitové binární číslo, jehož hodnota u prvního modulu (tj. úplně vlevo z předního pohledu na panel) činí 000_2 a je tedy možné takto zapojit až 8 modulů.

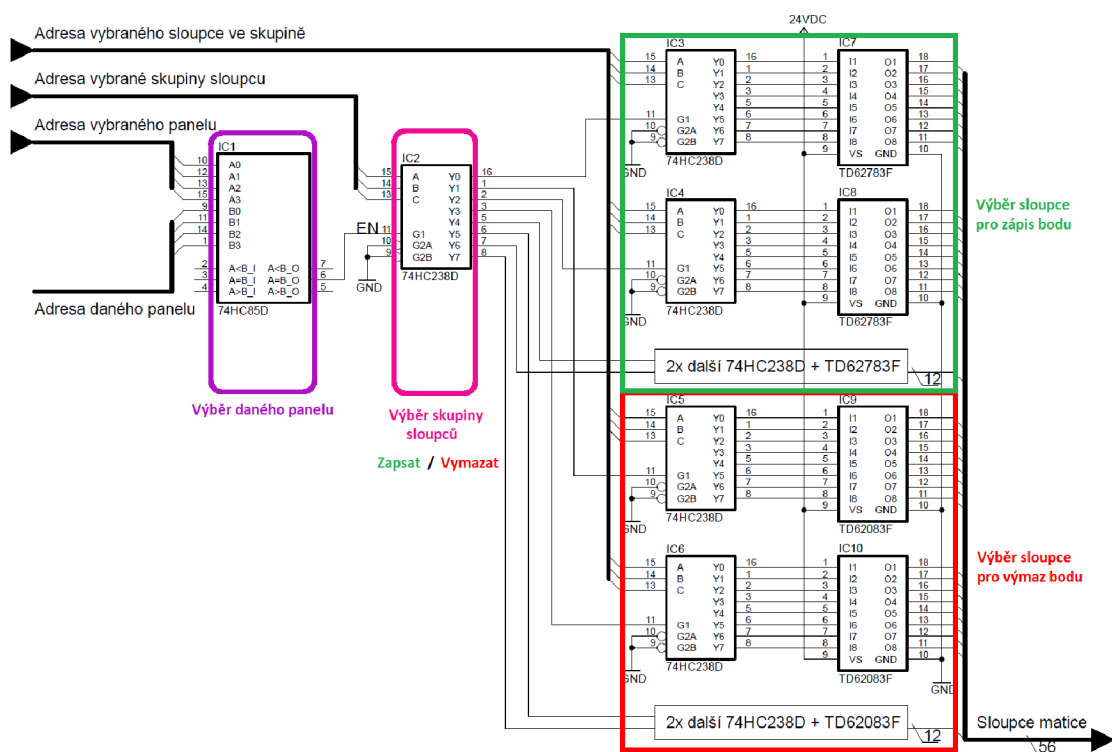
5) V páté skupině jsou přivedeny stejnosměrná **napájecí napětí** 5 V, 12 V, 24 V.



Obr. 2.3 Popis kontaktů propojovacího 50pinového konektoru modulů BS110

Krom konektorů se na zadní straně ještě nachází elektronický obvod. Ten se stará o správnou adresaci modulů a řízení napětí sloupců zobrazovací matice.

Za nevýhodu tohoto zapojení lze považovat fakt, že v jeden okamžik lze přistupovat pouze k jednomu sloupci displeje. To však v případě tohoto typu displeje nehraje ve výsledku zase tak velkou roli, protože napájecí napětí 24 V poskytuje po připojení na elektromagnetické cívky terčků dostatek energie i za poměrně krátkou dobu. Doba nutná pro otočení jednoho terčíku se tak pohybu v řádu milisekund, což se považuje za dostatečné pro informační tabule této velikosti. To i vzhledem k tomu, že doba potřebná pro změnu zobrazení může být závislá na počtu změn/otočení terčků (protože ty probíhají za sebou) a málokdy dochází ke změně všech bodů displeje najednou.



Obr. 2.4 Principiální schéma funkce výběru sloupce modulu displeje

Samotný výběr jednoho sloupce spočívá v povolení výstupu následujícího kaskádně řazeného logického obvodu signálem „enable“ označovaného EN. (v obvodu na obr. 2.13 jsou tyto piny označeny G1) Toto kaskádní zapojení se skládá z následujících integrovaných obvodů:

- A. **Jedna** binární 4bitová plná sčítanka nesoucí označení **HC283**. Ta se stará o správné řazení adresy modulů. Z pinů pravého 50pinového konektoru sloužících pro předávání adresy získá 3bitovou hodnotu. Ta u krajního (levého) modulu činí 000_2 . Tuto hodnotu inkrementuje o 1 a hodnotu 001_2 poskytne následujícímu modulu, připojenému na levý 50pinový konektor. Díky tomu je adresace modulů vždy závislá pouze na jejich pozici.
- B. **Jeden** 4bitový komparátor s označením **74HC85**, který se stará o výběr daného modulu displeje. V tomto případě vždy jednoho z pěti osazených modulů.
- C. **Jeden** 8bitovým dekodér označený **74HC238D**. Ten následně vybere, zda se bude na požadovaném sloupci zapisovat či mazat a do které ze čtyř skupin požadovaný sloupec patří. Toho je docíleno rozdělením na dvě čtveřice čili celkem 8 možností výběru.

- D. **Dvě čtveřice** na sobě nezávislých 8bitových dekodérů se stejným označením jako předchozí, a to **74HC238D**. Ty už provádí konečný výběr jednoho z osmi sloupců nacházejících se v dané skupině.
- E. Na konci opět následují **dvě čtveřice** 8kanálových budících obvodů. Ty již pouze posilují výstupy předcházejících nevykonných dekodérů. Jedna čtveřice integrovaných obvodů se stará o připojení cívek ke kladnému napájecímu napětí 24 V. Každý z nich nese označení **TD62783F**. Integrované obvody druhé čtveřice jsou označeny **TD62083F** a umožňují připojení cívek k nulovému potenciálu zdroje 24 V. Díky tomu je umožněn tok proudu cívkou oběma směry.

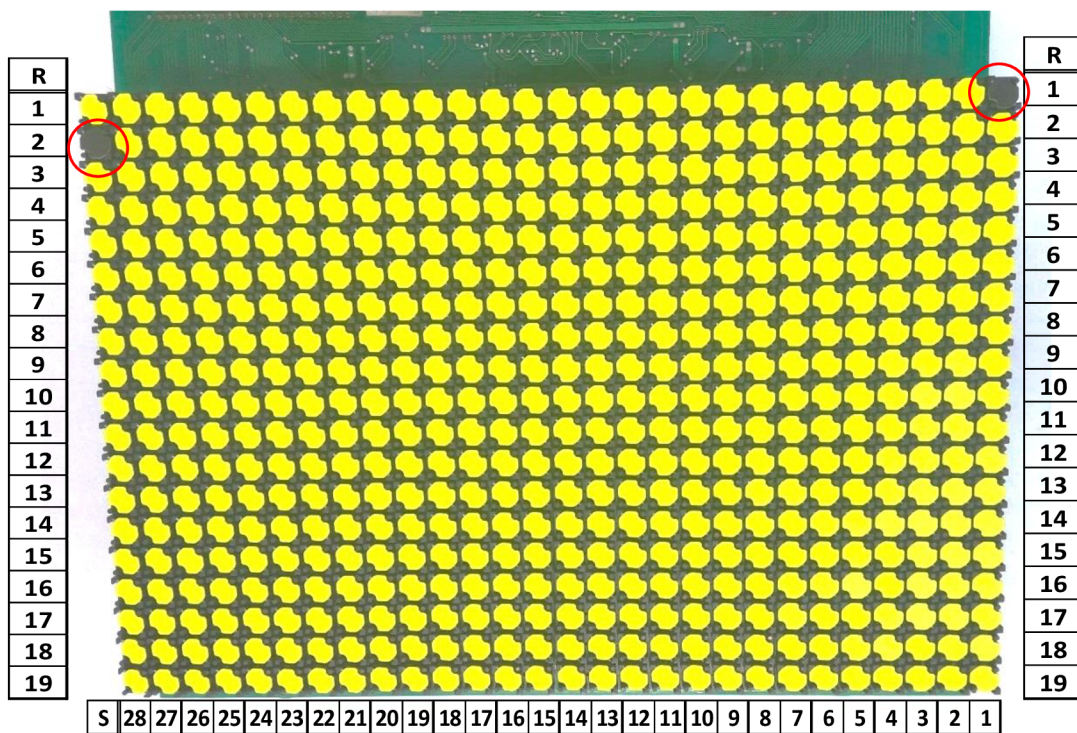
Maximální proud tekoucí ovládací cívkou displeje je tedy omezen vlastní impedancí cívky při přivedeném ovládacím napětí a maximální přípustnou hodnotu proudu výše zmíněných obvodu TD6278F a TD62083F. U obou obvodů je hodnota shodná, a to 500 mA. Ta sice převyšuje doporučenou hodnotu proudu tekoucí ovládací cívkou, která má mít hodnotu 350 mA, ale ani zdaleka se neblíží ustálené hodnotě proudu. Každá cívka má hodnotu odporu zhruba 16 Ω , což při hodnotě ovládacího napětí 24 V a úbytku 0,6 V na diodě zapojené v sérii dává proud:

$$I = \frac{U}{R} \quad 1$$

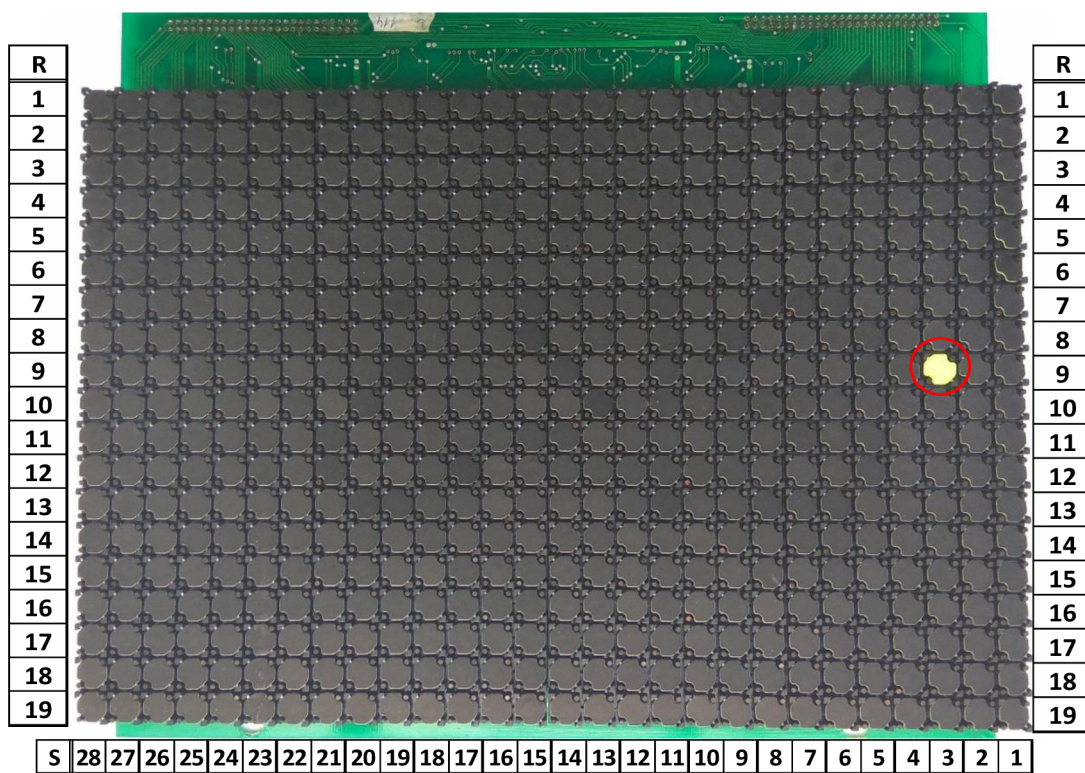
$$I = \frac{24 - 0,6}{16} \doteq 1,46 \text{ A}$$

Z toho důvodu je nejen možné zvládnout otáčení terčíků za jednotky milisekund, ale nezbytně nutné omezit dobu průchodu proudu pro omezení jeho velikosti na hodnotu menší jak 200 mA, danou maximálními dovolenými proudy diod BAV99. [27]

Jelikož, jak již bylo zmíněno, se celý displej skládá z pětice stejných zobrazovacích modulů, stačilo pro lepší pochopení funkcí řídicího obvodu vyjmout pouze jeden z nich a ten nastudovat. Modul byl následně podroben optické inspekci stávajících elektronických komponent a za využití měřiče kontinuity obvodů (multimetru) bylo sestaveno jeho schéma zapojení. Mnou takto zrekonstruované schéma zapojení řídicích obvodů každého z modulů je možné vidět v příloze A. Zrekonstruované schéma zapojení modulu BS110. Z něj bylo pro zjednodušení navíc odvozeno tzv. principiální schéma zapojení. To je možné vidět na obr. 2.13. Jde o kaskádní zapojení několika logických obvodů, díky němuž lze ve výsledku docílit řízení 140 sloupců displeje pouze za pomoci 10 řídicích vodičů, jak je vidět na obr. 2.12.



Obr. 2.6 Ukázka vadný pixelů – žlutá strana



Obr. 2.5 Ukázka vadný pixelů – černá strana

2.1.2 Ověření funkčnosti

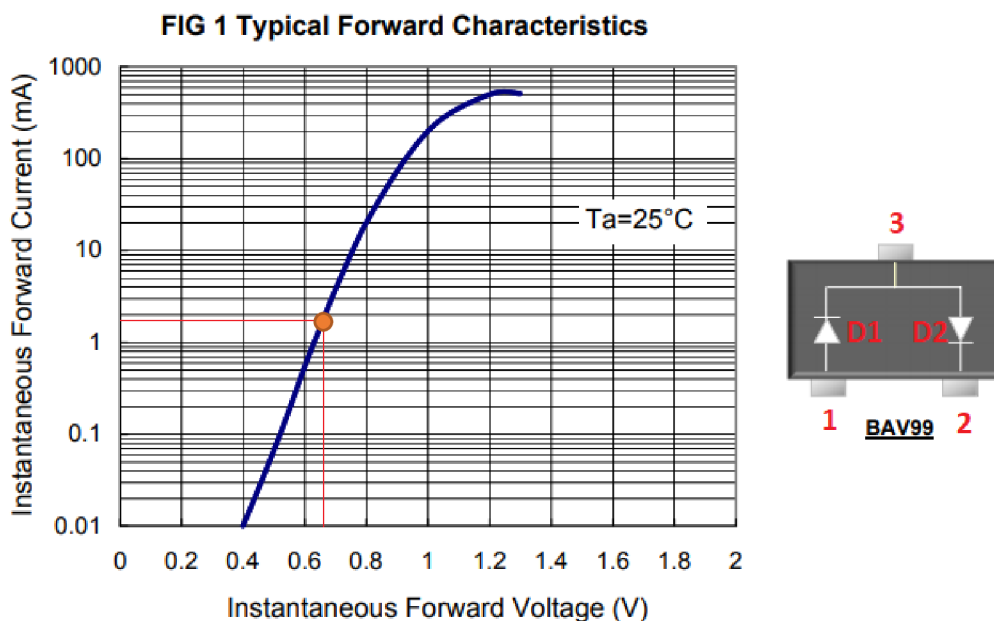
Pro ověření nabitých poznatků z analýzy zapojení následovalo krátké testování, při němž bylo využito dnes jednoduše dostupného jednodeskového mikrokontrolérového počítače Arduino MEGA 2560 společně s několika moduly řídicích můstků označených L298n. Testování probíhalo při výrazně sníženém napětí přiváděném na samotné ovládací cívky modulu displeje. Jelikož změřená hodnota odporu činila zhruba 16Ω , bylo ovládací napětí omezeno z 24 V na hodnotu 7 V. I takto výrazně snížené napětí však dokázalo bez větších komplikací otáčet jednotlivými terčíky, a navíc při něm nehrozilo zničení řídicích obvodů i při trvalém připojení. Na závěr tohoto krátkého testování se potvrdily výše uvedený popis funkce a postupy řízení jednotlivých bodů.

Mnohem lepší možnosti testování přinesla až nově navržená a sestavená řídicí elektronika spolu se základy řídicího programového vybavení. Díky tomu se podařilo zjistit že na 1. modulu displeje se nachází 3 prokazatelně vadné pixely, jak je možné vidět na obr. 2.5 a obr. 2.6.

Bohužel z důvodů výjimečné situace a vládních opatření bylo toto ověření funkčnosti provedeno pouze na jedné části displeje, kterou si autor jednoduše odvezl domů. Lze však přepokládat, že případné závady na ostatních modulech budou mít totožný charakter. Analýze a následné opravě se věnuje další podkapitola.

2.1.3 Analýza a oprava zjištěných závad

Pro snadnější identifikaci poškozených zobrazovacích bodů jsou tyto v textu označeny podle jejich polohy, a to ve formátu **SxRy**, kde písmeno **S** označuje **sloupec**, písmeno **R** **řádek** a písmena **x** a **y** nahrazují jejich **hodnoty od počátku** vyznačené na zadní straně



Obr. 2.7 parametry diody BAV99 [27]

DPS. Body *S1R1* a *S28R2* shodně umožňovali pouze otočení tmavou stranou, zatímco bod *S3R9* pouze světlou stranou. Jelikož vadné body byly ve zobrazovací matici rozmístěny náhodně, bylo jasné, že vada nebude způsobena řídicími obvody. Ta by se totiž díky principu řízení jednotlivých bodů po řádcích a sloupcích projevila chybami v celém řádku či sloupci. Chyba tedy mohla být až za řídicími obvody, za kterými se nachází již pouze ochranné diody a samotná ovládací cívka. Každý bod vlastní v jednom pouzdře SOT-23 nesoucího označení A7 dvojici diod s názvem BAV99. Jejich hlavní vlastností je propouštět proud pouze jedním směrem, přičemž na nich vzniká úbytek napětí. Podle jejich katalogových listů [27] lze zjistit, že hodnota tohoto napětí v propustném směru činí v závislosti na velikosti procházejícího proudu od 0,4 V po asi 1,3 V. U dostupného multimetru UNI-T UT33D byl měřením hrubě stanoven měřicí proud asi 1,75 mA, a tak musí naměřené napětí odpovídat hodnotě přibližně 0,6 V až 0,8 V. V závěrném směru nesmí být možné zmiňovaným multimetrem úbytek naměřit, respektive zobrazí se informace OL, která značí že měřená hodnota je mimo rozsah multimetru.

Výsledky měření jsou uvedeny v tabulce 2.1, kde označení „Uf“ značí napětí v propustném směru a „Ur“ značí napětí v závěrném směru. Z výsledků a principu funkce popsaných v podkapitole 2.1.1 Analýza funkčnosti je jasné, že zjištěné vady, kterými je myšleno zneprůchodnění cest proudu, způsobují právě tyto diody. Ty totiž neumožňují průchod proudu ze sběrnice řádků do sběrnice sloupců pro body *S1R1* a *S28R2*. Pro bod *S3R9* neumožňuje průchod ve směru opačné. Z tohoto důvodu zůstávaly tyto body beze změn.

Závada	Označení bodu	Číslo diody	Typ napětí	Naměřené napětí [mV]
Pouze tmavá strana	S1R1	D1	Uf1	718
			Ur1	nezměřitelné
		D2	Uf2	nezměřitelné
			Ur2	nezměřitelné
Pouze tmavá strana	S28R2	D1	Uf1	720
			Ur1	nezměřitelné
		D2	Uf2	nezměřitelné
			Ur2	nezměřitelné
Pouze světlá strana	S3R9	D1	Uf1	1700
			Ur1	nezměřitelné
		D2	Uf2	713
			Ur2	nezměřitelné

Tab. 2.1 Tabulka naměřených hodnot úbytků napětí diod BAV99

Oprava této závady byla provedena výměnou poškozených diod BAV99. U bodů *S1R1* a *S28R2* se vadu podařilo odstranit.

U bodu S3R9 však i nadále přetrvávaly její příznaky. Nezbyvalo tedy než se zaměřit na samotnou cívku. Po jemném odstranění laku z jejích vývodů využil autor zmíněného multimetru a změřil hrubou hodnotu odporu cívky. Ta u náhodně vybraného vzorku cívek dosahovala hodnoty $R \cong 16 \Omega$. U bodu S3R9 byla ale pouze $R \cong 3 \Omega$. Z těchto výsledků lze usoudit, že byla narušena izolační bariéra mezi závity cívky. Ta bývá nejčastěji tvořena izolačním lakem či podobným izolačním nátěrem naneseným na měděném vodiči. Dle mého názoru k poruše došlo nejspíše při dlouhodobém připojení cívky ke zdroji napětí a následnému tepelnému namáhání.

Vezměme v úvahu, že maximální dovolený proud je dán nejslabší součástí obvodu, kterou jsou právě již zmíněné diody BAV99. U nich výrobce v katalogových listech uvádí maximální špičkový proud $I_{FRM} = 450mA$ a maximální střední hodnotu proudu $I_O = 200mA$. Tepelný ztrátový výkon na cívce v ustáleném čase je dán vztahem:

$$P = U \cdot I = R \cdot I^2 \quad 2$$

$$P \cong 18 \cdot 0,2^2 \cong 0,72W$$

Při trvalém průchodu maximální povolené střední hodnoty proudu diodami, nemůže dojít k poškození žádné z částí řídicí elektroniky, avšak teplo hromadící se na vinutí může způsobit degradaci až porušení izolace mezi jednotlivými závity vinutí. Výsledný zkrat snížil hodnotu odporu cívky, a protože byl panel s největší pravděpodobností napájen napěťovým zdrojem, právě snížený odpor zapříčinil zvýšení procházejícího proudu a následné zničení nejslabší součástky, a to právě mnou vyměněné diody S3R9.

Pro opravu této závady by bylo potřeba buď zakoupit nový „štrips“, nebo se pokusit opravit poškozené vinutí stávajícího. Protože ale tyto štripsy nejsou jednoduše k dostání, odpadá první varianta a na druhou variantu si autor z důvodů omezených prostředků a v domácích podmínkách netroufá.

Výsledkem je tedy jeden modul s jedním vadným pixelem a, jak už bylo jednou uvedeno, 4 moduly bez zjištěných závad. S největší pravděpodobností však budou mít případné závady stejnou příčinu, a to poškození diod jednotlivých bodů, popř. mezi závitový zkrat, či možná dokonce přerušenu ovládací cívku. Jak už bylo zmíněno výše, kdyby byl problém s logickými či koncovými obvody řízení sloupců, byla by stejná vada vidět pro všechny body v daném sloupci, tedy tvořila by na displeji vertikální čáry.

2.2 Napájecí zdroj

Kromě zmíněných zobrazovacích modulů byl u panelu k dispozici i funkční síťový napájecí zdroj, vyrobený pro požadavky předešlých závěrečných prací.

Vstupní napájecí napětí tohoto zdroje odpovídá napětí sítě užívané v ČR, a to 230V/50 Hz. Toto napětí je přiváděno skrze dvě vstupní tavné pojistky na dvojici snižujících transformátorů. Ty se starají nejen o snížení napětí, a to na 9V a

24V střídavých, ale i o galvanické oddělení od sítě. Na výstupu obou transformátorů se nachází Graetzův usměrňovací můstek a filtrační kondenzátory.

Výstup za usměrňovačem 9voltového transformátoru je přiváděn na první z dvojice spínaných snižujících měničů. Ten nese označení **LM2596T-5.0** a stará se o efektivní snížení výstupního napětí na hodnotu 5 V, která se využívá pro napájení logických obvodů modulů BS110.

Druhý z dvojice měničů je připojen za usměrňovač 24voltového transformátoru a nese označení **LM2575T-12**. Ten se stará taktéž o efektivní snížení výstupního napětí, a to na hodnotu 12 V potřebnou pro vstupní obvody modulů BS110. Paralelně k tomuto obvodu je připojen i lineární stabilizátor označený **L78S24CV**, který snižuje napětí pro ovládání cívek modulů BS110 na hodnotu 24 V.

Výstupy všech tří napěťových zdrojů jsou taktéž opatřeny tavnými pojistkami chránícími části zdroje proti zničení zkratem výstupních vodičů.

Zdroj je funkční, nenesé žádné známky závad a není jej tudíž nutné opravovat. Pro potřeby práce je dostačující.

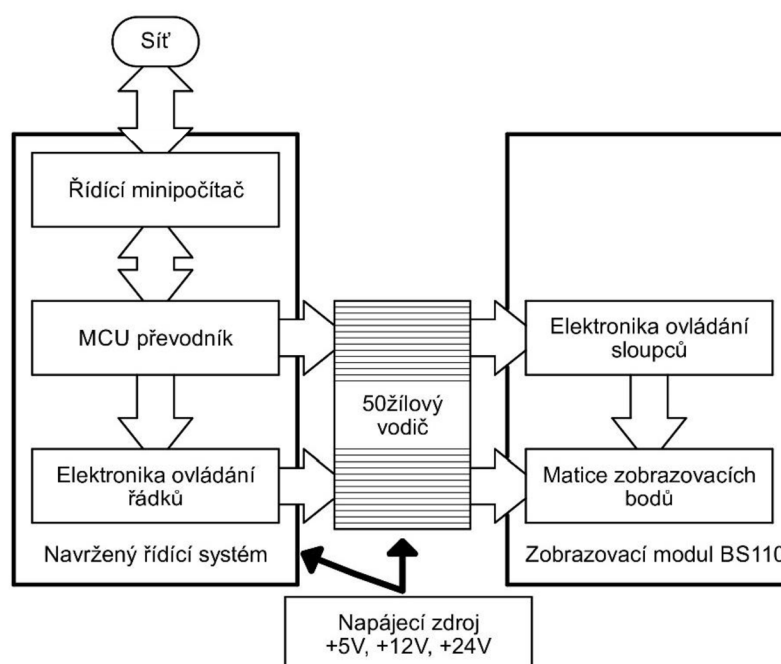
3. NÁVRH ZAŘÍZENÍ

Podle požadavků daných zadáním práce má vzniklý mikrokontrolerový řídicí systém být schopen řídit zobrazovací jednotku informačního panelu BS110 a umožňovat komunikaci s okolím skrze Ethernetové a bezdrátové rozhraní. Nahradí tak původní řídicí jednotku, která byla určena k fungování uvnitř palubního informačního systému veřejné dopravy a umožňovala komunikaci pouze skrze sběrnici IPIS.

Návrh začíná od samotného ovládání displeje. Displej, jak bylo již několikrát zmíněno, obsahuje vlastní elektroniku pro řízení napětí jednotlivých sloupců. Co naopak postrádá a při návrhu bylo prvním úkolem, je chybějící elektronika pro řízení napětí na 19 řádkových pinech vyvedených pomocí propojovacího 50žilového kabelu.

Další částí, o kterou je systém potřeba doplnit, je bezesporu řídicí jednotku. Ta má za úkol nejen řídit obvody samotného displeje, ale i komunikovat skrze požadovaná rozhraní. V tomto případě si lze buď takovýto řídicí systém obsahující kontrolér sestavit zcela sám z elektronických součástek, nebo využít některých z již dostupných řešení. Podobně to platí i při hledání požadovaných vlastností jako například možnosti Ethernetové a bezdrátové komunikace. Lze zvolit kontrolér, který sám tyto komunikace neumožňuje a rozšířit jej o externí obvody či moduly.

Pro urychlení návrhu je však výrazně vhodnější varianta, kdy se zvolí již známý a osvědčený systém, který sám požadované vlastnosti již zahrnuje. I z tohoto důvodu byl tedy vybrán minipočítač Raspberry Pi. Tomu sekunduje ještě jednoduchý mikrokontrolér ve funkci převodníku. Ten má za úkol přijímat informace od nadřazeného minipočítače a

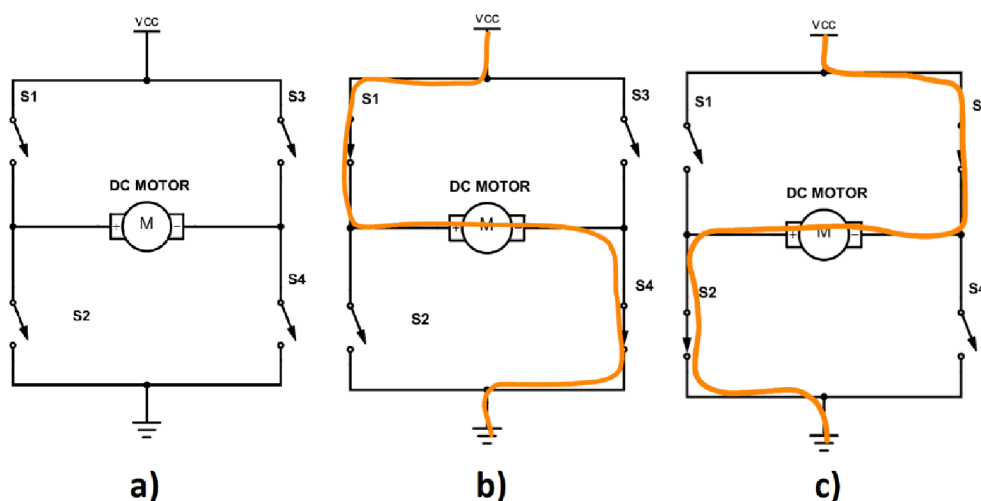


Obr. 3.1 Blokové schéma návrhu zařízení

ty následně skrze ovládací obvody displeje převést na požadovaný obraz. Výsledná koncepce navrženého systému je vidět na obr. 3.1.

3.1 Elektronika řízení řádků

Jak již bylo zmíněno, elektronika pro ovládání sloupců je již součástí zobrazovacích modulů, zbývá vyřešit ovládání řádků displeje. Požadavkem pro tento obvod je relativně rychlé (v řádech milisekund) připojování cívek vinutí na kladné napětí +24 V či zem označovanou GND. Jde tedy o spínání indukivní zátěže, stejně jako je tomu například při řízení elektrických motorů. Při tom se v praxi využívají obvody známé jako **řídící H-můstky**. Jak je vidět na obr. 3.2 skládají se ze čtveřice spínačů. V praxi jde nejčastěji o



Obr. 3.2 Zapojení a princip funkce H-můstku [17]

čtveřici bipolárních tranzistorů, např. v Darlingtonově zapojení. Každý H-můstek lze rozdělit na dvojici tzv. „půl-můstků“, kdy každý z nich je připojen na jednu stranu vnutí motoru a každý lze ovládat zvlášť, a tedy připojit skrze spínač S1(S3) na kladné či skrze spínač S2(S4) na záporné napájecí napětí. [9]

Tyto řídicí obvody lze nalézt v mnoha variantách jako již hotová řešení v podobě integrovaných obvodů. **Požadavky na řídicí obvod pro řádky displeje** jsou dány zobrazovacími moduly a vypadají následovně:

Proud tekoucí cívkami do **500 mA**

Napětí na cívkách okolo **24 V**

Spínací frekvence **>1kHz**

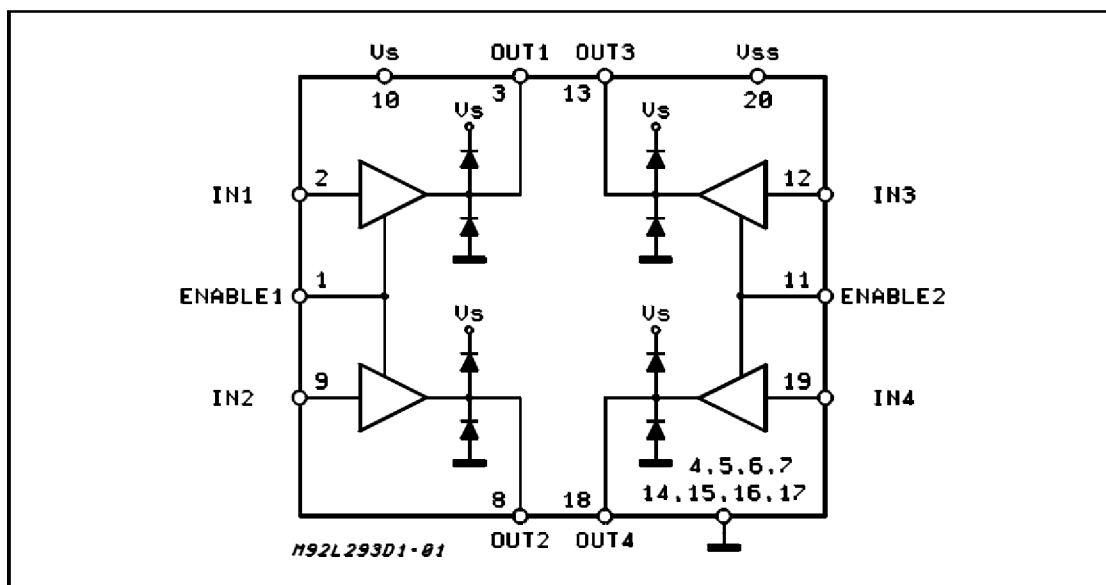
Parametry		Požadavky	L293D	L298n	L9110
Cena	[Kč]	není	56	78	29
Výstupní proud	[mA]	>500	<600	<2000	<800
Špičkový proud	[mA]	není	<1200	<4000	<1500
Ovládací napětí	[V]	>24	<36	<46	<12
Spínací frekvence	[kHz]	>1	<5	<25	<40
Pouzdra		není	SO20, DIP16	SO20, Multiwatt15	SOP8, DIP8
Další parametry:		není	2 H-můstky v pouzdře	2 H-můstky v pouzdře	1 H-můstek v pouzdře
			Integrované ochranné diody	není	Integrované ochranné diody
			Vstup EN	Vstup EN	není

Pozn. Uvedené ceny jsou z internetového obchodu GME.cz k 5.1.2020

Tab. 3.1 Porovnání parametrů při výběru H-můstku

Mezi dostupnými obvody jsou v tab. 3.1 porovnány 3 vybrané. Obvod **L9110** nespĺňuje požadavek na řídicí napětí a neumožňuje výběr aktivního H-můstku signálem EN. Obvod **L298n** sice všechny požadavky splnil, je však pro danou aplikaci zbytečně výkonný, nejdražší z trojice a neobsahuje ochranné diody. Výběr tedy zůstal na obvodu s označením **L293D**. Ten s dostatečnou rezervou splňuje výše uvedené požadavky. Uvnitř obvodu se nachází 2 H-můstky. Díky tomu je pro ovládání 19 řádků potřeba pouze 5 kusů tohoto obvodu. Každý v sobě pro ochranu nese již integrovanou dvojici diod, díky níž lze opět zjednodušit výsledné zapojení. (Diody slouží jako ochrana proti napěťovým špičkám generovaných při odpojování indukčních zátěží, vlivem elektromagnetické indukce). Kromě toho obvod obsahuje funkci povolování výstupu signálem EN společným

BLOCK DIAGRAM



Obr. 3.3 Blokový diagram vnitřního zapojení obvodu L293D

vždy pro jeden H-můstek, jak je vidět v tab. 3.2. Obě tyto vlastnosti je možné vidět na blokovém diagramu na obr. 3.3.

Od původního plánu, řídit každý jeden řádek displeje pomocí jednoho pinu MCU, bylo nakonec upuštěno. Jak již bylo uvedeno v kapitole 2.3.1, nemá velký smysl

TRUTH TABLE (one channel)

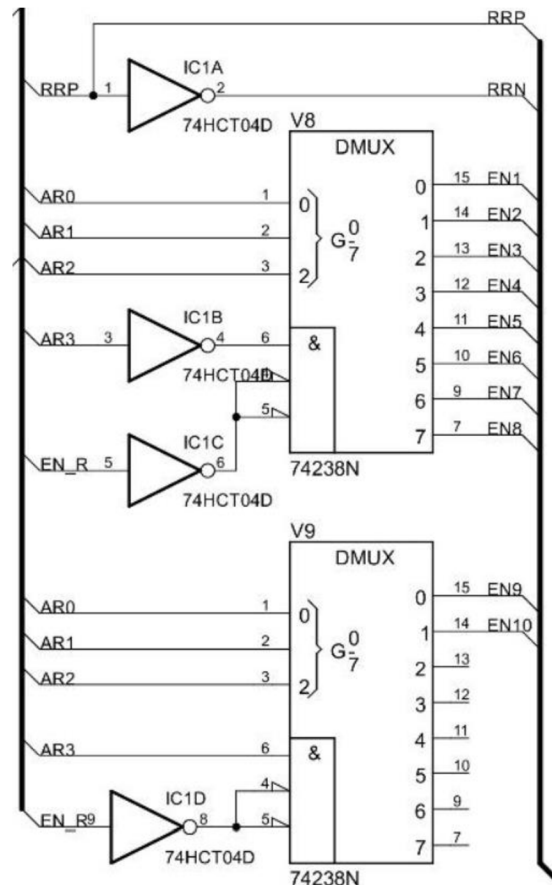
Input	Enable (*)	Output
H	H	H
L	H	L
H	L	Z
L	L	Z

Z = High output impedance

(*) Relative to the considered channel

Tab. 3.2 Hodnoty výstupu obvodu L293D v závislosti na signálu EN

obsazovat 19 vývodů řídicího MCU, protože již připojení jediné ovládací cívky displeje na napájecí napětí dokáže vyvolat maximální povolený proud protékající řídicími obvody jednotlivých sloupců. Z tohoto důvodu došlo k rozšíření původního zapojení, kde MCU měl za úkol řídit všech 19 vstupů obvodů L293D napřímo. Namísto toho byl přidán obvod pro výběr 1 řádku z 19. Zapojení obvodu je možné vidět na obr. 3.4. Využívá vždy dvojice povolovacích signálů, které se na L293D nacházejí. Základem pro něj jsou dva stejné binární dekodéry označované jako „1 z 8“, které jsou již použity pro výběr sloupců na samotných modulech displeje. Jde tedy o dvojici obvodů **74HC238D**. Ty s pomocí jednoho z šesti hradel NOT v pouzdře s označením **74HCT04D** tvoří ve výsledku dekodér označovaný jako „1 z 16“. Tím je zaručen výběr vždy jednoho z 10 H-můstků pomocí 4místné adresy neboli čtveřice pinů MCU. Pro výběr jediného řádku je potřeba provést zápis pomocí jednoho signálu, a to jen na jednu polovinu již vybraného H-můstku. Toho je docíleno pomocí druhého hradla NOT tak, že vstupní signál druhé poloviny H-můstku je pomocí něj znegován. Díky tomu dochází vždy k ovládní pouze jednoho řádku. Pro řízení řádků došlo tedy k redukci z **19** signálů na 4 nosící adresu dvojice řádků, 1



Obr. 3.4 Obvod pro výběr 1 z 19 řádků

nesoucí informaci o tom, zda se bude zapisovat či mazat a 1 povolovací, celkem tedy 6 vodičů.

	Zápis (+24V)	Výmaz (GND)
Lichý řádek	1 (H)	0 (L)
Sudý řádek	0 (L)	1 (H)

Tab. 3.3 Hodnota řídicího signálu pro výběr řádku

Takže pokud zapisujeme na bod displeje ležící například na 1. řádku připojeném k 1. polovině H-můstku zapíšeme na řídicí pin MCU kladnou hodnotu (tj. HIGH) což zapříčiní připojení k +24 V. Na 2. polovinu (tj. 2. řádek) bude v tomto případě přivedena záporná hodnota, tj. GND (0 V). Proud tekoucí z +24 V do připojeného sloupce (0 V) vyvolá v 1. řádku změnu zobrazovaného bodu, zatím co mezi dvěma stejnými potenciály ve 2. řádku k žádné změně nedojde. Pro stejný zápis na 2. řádek je postupováno stejně s tím rozdílem, že řídicí signál bude znegován, jak je vidět v tab. 3.3. Pro možnost synchronizovaného spuštění je obvod výběru řádku vybaven o povolovací signál EN přiváděný na oba dekodéry pomocí zbylých hradel NOT. Celé zapojení obvodu je vidět v příloze B. Návrhu řídicí elektroniky verze 1.

3.2 Řídicí minipočítač

Jelikož je v zadání práce uveden požadavek na Ethernetové i bezdrátové propojení s okolím, byl po delší rešerši nakonec, jako nejvyšší řídicí prvek, vybrán dnes velmi rozšířený minipočítač označovaný jako Raspberry pi, a to konkrétně ve verzi 3 model B+.

Parametry	Rapsberry Pi 3 model B+
CPU	ARM Cortex-A53 1.4GHz
RAM	1GB SRAM
Wi-Fi	2.4GHz and 5GHz
Ethernet	300Mbps
PoE	Yes
Bluetoooh	4.2
Cena	1 149 Kč

Cena je převzatá z internetového obchodu GME.cz k 5.1.2020

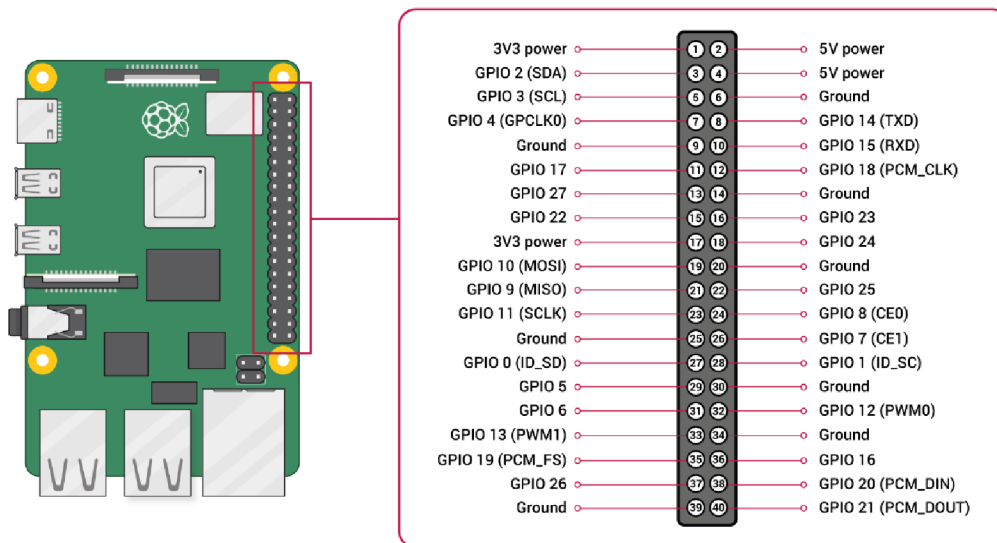
Tab. 3.4 Tabulka základních parametrů použitého minipočítače [18]

Jde o poměrně levný ale za to velmi dobře vybavený počítač velikosti kreditní karty. Jeho základní parametry jsou uvedeny v tab. 3.4.

Jak je z tabulky vidět splňuje všechny zadané požadavky, neboť má nejen Ethernetové připojení do rychlosti až 300Mbps, ale i obě dnes nejrozšířenější používaná pásma Wi-Fi

připojení, ba dokonce navíc i Bluetooth 4.2. Jeho výhodou oproti mikrokontrolerům je přítomnost vlastního operačního systému. Ten velmi zjednodušuje komunikaci přes internet a umožňuje výběr z mnoha programovacích jazyků a knihoven.

Jako interní paměť standardně využívá buď Micro SDHC karet, či USB flash disků. Pro propojení s hardwarem má kromě standardních počítačových sběrnic, jako jsou USB 2.0, Ethernet, HDMI, i 40pinový konektor označovaný jako GPIO. (General-purpose input/output) Na něm se nachází další komunikační sběrnice jako jsou SPI, I²C a UART často užívané u jednoduchých mikrokontrolerů.



Obr. 3.5 Označení pinů Raspberry Pi konektoru GPIO [19]

3.3 Pomocný MCU převodník

Jako mezičlánek, mezi řídicím minipočítačem s běžícím operačním systémem a hardwarovou logikou ovládacích obvodů displeje, byl vybrán 8bitový mikrokontrolér. Ten bude mít za úkol zobrazovat na panelu znaky, které z nadřazeného minipočítače obdrží. Pro přenos dat mezi oběma zmíněnými jednotkami je nejjednodušší volbou jednoduchá asynchronní komunikace, kterou jsou již obě komunikační strany vybaveny, označovaná jako **UART**. Ta pro komunikaci využívá pouze dvojici vodičů označovaných jako TX (Odesílaná data) a RX (přijímaná data).

Hlavními požadavky na mikrokontrolér převodníku jsou dostatečný **počet výstupních pinů** a paměť pro základní zobrazované znaky. Z běžně dostupných 8bitových mikrokontrolerů byl nakonec vybrán již tradiční mikrokontroler od výrobce Microchip s označením ATMEGA32 v pouzdře DIL40. Ten má čtveřici 8bitových vstupně výstupních portů, taktéž označovaných jako GPIO. Celkem tedy 32 dostupných

GPIO pinů. Tento počet dostačuje nejen pro řídicí elektroniku V2 i pro původně zamýšlenou variantu řízení 19 řádků přímo z MCU pinů:

$$19 \text{ pinů pro řádky} + 1EN_{\text{řádky}} + 10 \text{ pinů pro sloupce} + 2 \text{ piny pro UART} = 32$$

Pro vylepšenou variantu řízení řádků displeje, kdy se množství řídicích pinů snížil z 20 na 6, je možné zbylých 14 pinů využít při rozšiřování funkcí informačního panelu, jako je například doplnění panelu o vlastní přisvětlovací jednotku reagující na intenzitu okolního světla.

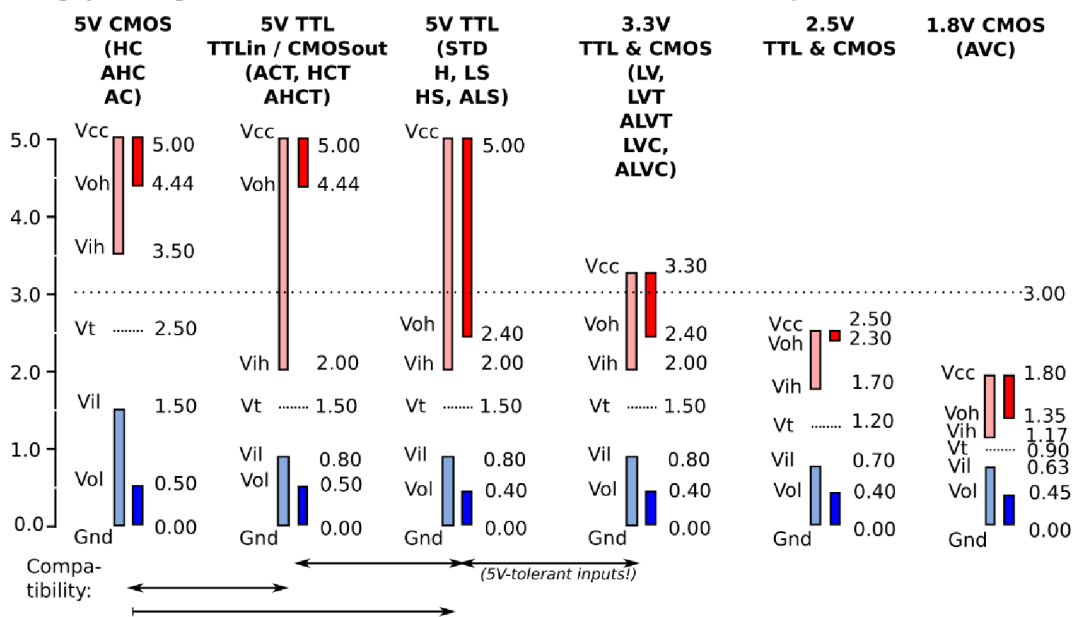
3.3.1 Převod napět'ových úrovní

Vybraný řídicí minipočítač Raspberry Pi 3 B+ využívá na svých GPIO pinech 3,3 V logiky, kdežto zobrazovací moduly BS110 dosahují logické úrovně 1 při napětí 5 V. Z tohoto důvodu je nutné tyto napětí vzájemně přizpůsobit.

ATmega32 (5V)		Porovnání	Raspberry Pi (3,3V)	
$U_{IH,min}[V]$	3	\leq	3	$U_{OH,min}[V]$
$U_{IL,max}[V]$	1	$>$	0,14	$U_{OL,max}[V]$

Tab. 3.5 Porovnání napět'ových úrovní při přenosu z Raspberry Pi do ATmega32

V případě přivádění napětí 3,3 V na piny 5voltové logiky nejde ve výsledku o komplikaci, neboť jak je vidět na obr. 3.6 je toto propojení tzv. tolerantní. Napět'ová úroveň logické hodnoty 1 totiž začíná již od napětí 2 V. Nicméně dokumentace samotného mikrokontroléru uvádí minimální hodnotu napětí pro logickou hodnotu 1 pro všechny piny (s výjimkou pinu XTAL1 a RESET) jako 0,6násobek napájecího napětí což pro napájecí napětí 5 V dává hodnotu 3,0 V. Z tohoto důvodu je sestavena tabulka 3.5.



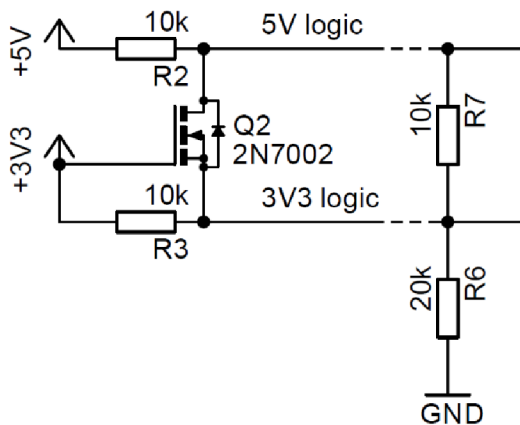
Data source: EETimes, A brief recap of popular logic standards (Mark Pearson, Maxi)

Obr. 3.6 Hodnoty logických napět'ových úrovní [20]

V ní je vidět porovnání úrovní napětí potřebných pro správný přenos dat z Raspberry Pi do mikrokontroléru ATmega32. [18]

V opačném směru přenosu dat, a to z ATmega32 do Raspberry Pi, by již komplikace nastaly, protože přivedené napětí 5 V by velice pravděpodobně způsobilo zničení 3,3voltových vstupních obvodů Raspberry Pi.

Pro případ posunu logických úrovní se v praxi využívá tzv. napěťových převodníků. (voltage level shifter/converter) [21] Toto řešení však lze pro tento jednoduchý případ nahradit obyčejným napěťovým děličem složeným z dvojice za sebou zapojených rezistorů. Porovnání úrovní napětí mezi děličem připojeným na výstup mikrokontroléru ATmega32 a Raspberry Pi je v tabulce 3.6. Z obou tabulek je vidět, že **minimální** hodnoty rozsahů obou úrovní napětí pro logickou hodnotu **1** (tj. první řádek) jsou na **výstupu vyšší** než na **vstupu**. Naopak pro logickou hodnotu **0** jsou **maximální** hodnoty rozsahů (tj. druhý řádek) obou úrovní napětí na **výstupu nižší** než na **vstupu**. Díky tomu je zaručena oboustranná tolerance napětí. Způsob zapojení převodníku je na obr. 3.7.



Obr. 3.7 Příklady zapojení napěťového

Jelikož se systém skládá ze tří celků, z nichž krajní dva mají rozdílné hodnoty napětí

ATmega32 (5V) napětí za děličem 2/3		Porovnání	Raspberry Pi (3,3V)	
$U_{OH,min}[V]$	2,8	>	1,6	$U_{IH,min}[V]$
$U_{OL,max}[V]$	0,47	<	0,9	$U_{IL,max}[V]$

Tab. 3.6 Porovnání napěťových úrovní při přenosu z ATmega32 do Raspberry Pi

logických úrovní a třetí celek (MCU převodník), který dva zmíněné spojuje, dokáže pracovat s každým z nich, vzniká zde možnost volby. Za nejlepší řešení je považováno, onu pomyslnou dělicí čáru mezi 3,3voltovou a 5voltovou logikou vytvořit mezi řídicím minipočítačem a MCU převodníkem, a to, protože je zde díky komunikaci typu UART zapotřebí pouze 2 vodičů, a nevzniká tak potřeba dalšího napájecího zdroje s napětím 3,3V.

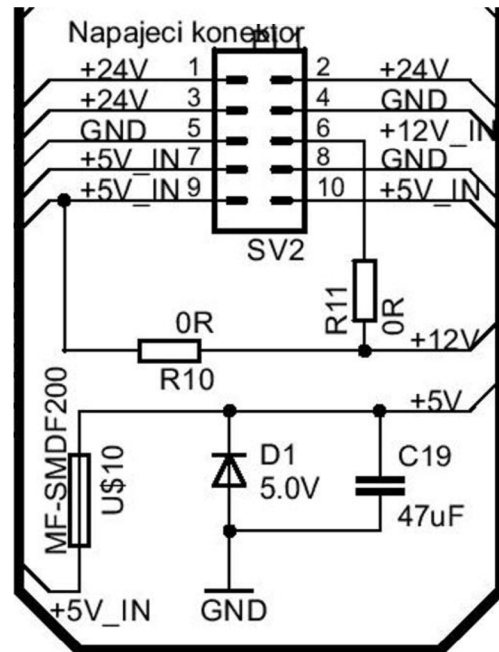
3.3.2 Rozšiřující konektor

Pro možnosti využití zbylých pinů mikrokontroleru bylo zařízení rozšířeno o jednoduchý 20pinový konektor JP5 označený jako „Expansion con“. Ten obsahuje krom zmíněných pinů i všechny napájecí větve. Lze jej například využít při zobrazování stavu MCU

převodníku skrze LED.

3.4 Napájení, ochrana proti nadproudu a přepětí 5V větve

Pro zařízení je vhodné, aby jeho napájení bylo přiváděno prostřednictvím pouze jediného kabelu s jedním konektorem. Tento konektor je ve schématu zapojení označen SV2 a přivádí do zařízení napětí 5 V, 12 V a 24 V a společný nulový potenciál označovaný GND. Elektronika samotné řídicí desky zařízení využívá pro své napájení napětí o hodnotě 5 V a řídí sběrnici řádků displejů prostřednictvím napětí 24 V. Všechny napájecí větve jsou prostřednictvím 50 pinového konektoru PL1 připojeny k zobrazovacím modulům. Při rozboru stávající elektroniky modulů displeje nebylo úplně jasné, z jakého důvodu jsou všechny signálové cesty připojeny přes rezistorové pole na napětí 12 V, tvořící tzv. „pullup“. Pro možnost vynechání tohoto zdroje a záměnu za 5 V byly přidány do obvodu 2 nulové rezistory R10 a R11, jak je možné vidět na obr. 3.8.



Obr. 3.8 Schéma zapojení napájení

Raspberry Pi však v praxi bývá často napájen přímo prostřednictvím micro USB konektoru. Za ním se totiž nachází vratná pojistka, ve funkci ochrany proti nadproudu, a dioda pro ochranu proti přepětí. Raspberry Pi lze napájet i prostřednictvím GPIO sběrnice na kterou jsou vyvedeny obě jeho napětí, a to 5 V a 3,3 V. Tyto piny jsou však určeny pro napájení připojených zařízení, a nikoliv samotného Raspberry Pi, protože obchází zmíněné ochrany.

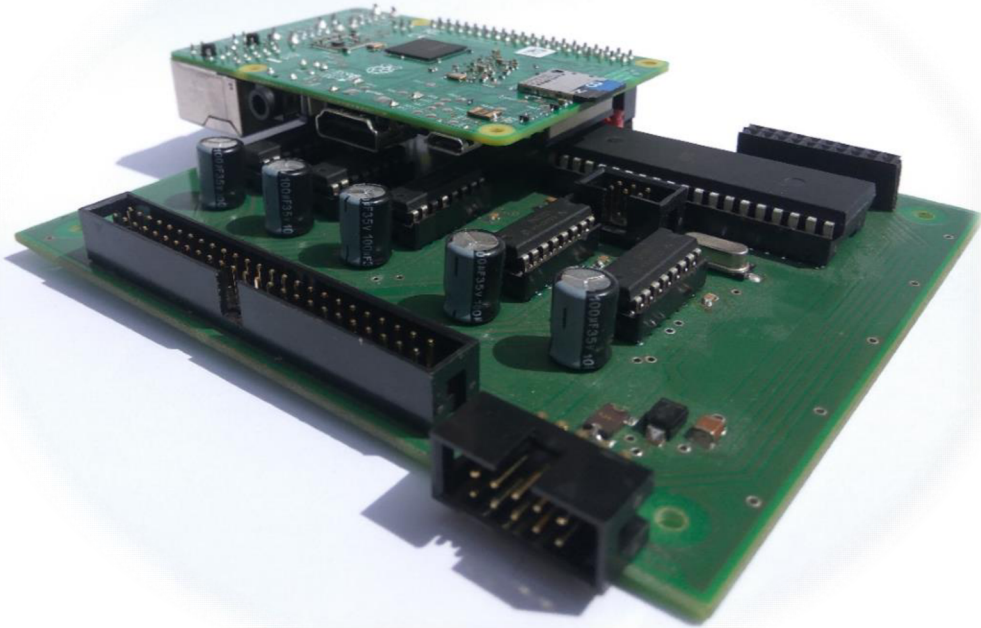
Z těchto důvodů je 5 V větev doplněna o obě zmíněné ochrany. Vratná pojistka s označením MF-MSMF250/16X je podobná té, která je na samotném Raspberry Pi. Její nejvyšší pracovní proud splňuje hodnotu nejvyššího odběru Raspberry Pi a činí 2,5 A. K samotnému odpojení dochází při proudu 5 A. O ochranu proti přepětí se stará ochranná dioda označená SMBJ5.0A, která byla vybrána opět podle samotného Raspberry Pi. Do hodnoty napětí 5 V ochranná dioda nevede proud vůbec a nad hodnotu 6,4 V potlačuje veškeré přepětí.

3.5 Sestavení návrhu a výroba HW

Pro jednodušší práci s prototypem byla vybrána DIP pouzdra hlavních součástí osazených

v patičkách. Díky tomu je lze jednoduše vyjmout a nahradit.

Jelikož má Raspberry Pi vyvedenou komunikaci i napájení na pinech GPIO sběrnice, je s ním DPS navrženého zařízení skrze tyto piny spojena pomocí speciálně vyvýšeného 40pinového konektoru určeného pro Raspberry Pi 3B+ v tzv sendvičové sestavě jak je vidět na obr 3.9.



Obr. 3.9 Výsledný vzhled vyrobeného HW

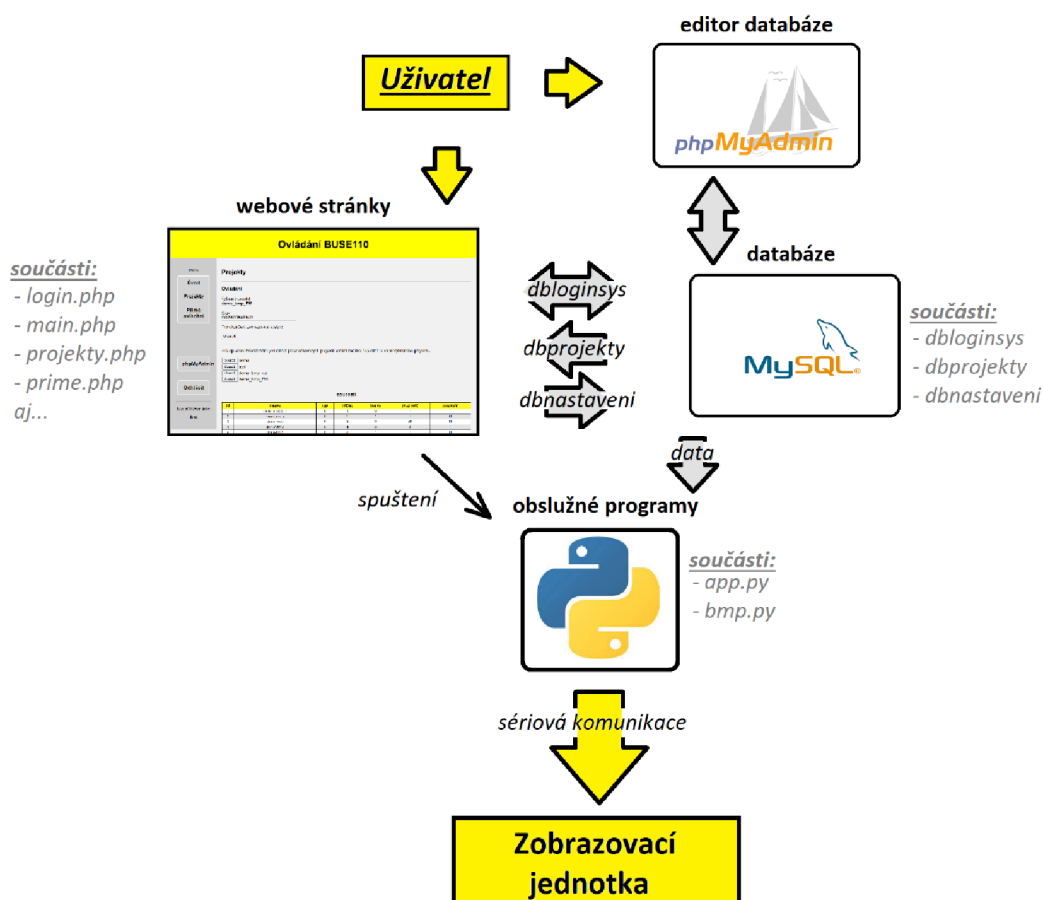
Pro takto navržené zařízení byla vytvořena technická dokumentace v programu EAGLE 7.2.0. Ta se sestává ze schématu a desky plošných spojů, které se nachází v přílohách 3, 4 a 5. Takto získané soubory byly předány odborné firmě k výrobě desek plošných spojů. Z technické dokumentace byl také sestaven seznam součástek tzv „BOM“, na jehož základě byly objednány jednotlivé součástky.

Po získání všech potřebných částí byl obvod postupně pájen pomocí standartní pájecí stanice a ožívován pomocí pro mě dostupného napěťového zdroje s omezením proudu. Důležité je totiž při ožívování prvního prototypu výrobku nic neuspěchat. Je přímo nevhodné celé zařízení sestavit, připojit na tzv. „tvrdý“ napěťový zdroj a čekat co se stane. Často se totiž může stát, že byl prototyp navržen s chybou, a to potom v nejlepším případě nic nefunguje a v nejhorším případě se objeví oblaka dýmu. Pro postupné ověřování funkcí jednotlivých částí obvodu bylo využito základů výsledného SW vybavení MCU, které bude popsáno v následující kapitole.

4. NÁVRH SW VYBAVENÍ

Další důležitou součástí zadání je požadavek na vhodné programové vybavení. Jelikož je zařízení rozděleno na dva programovatelné celky, kterými jsou mikrokontrolér a minipočítač, bylo potřeba vytvořit dvou oddělených programových výbav.

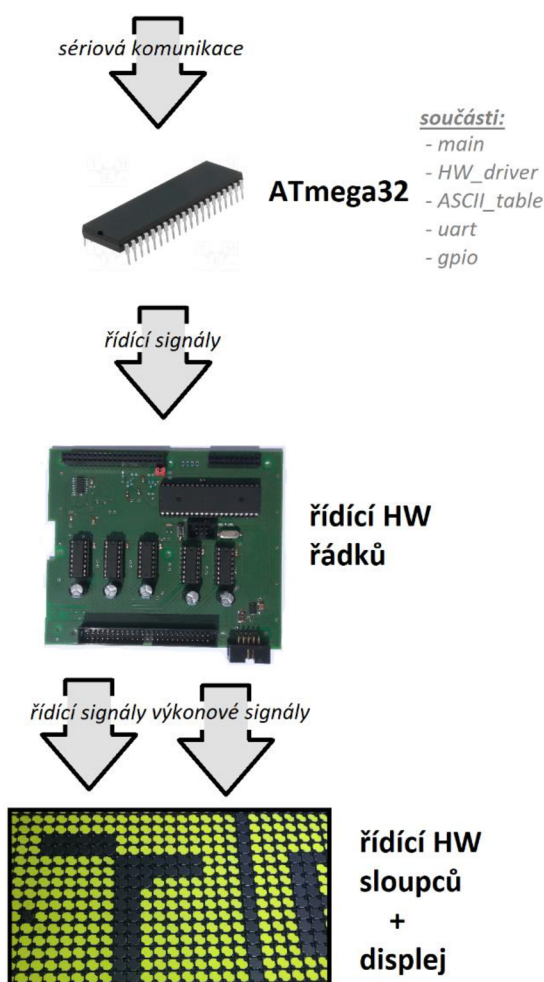
Z pohledu úvah při návrhu koncepce se autor nejprve zaměřil na možnosti mini počítače Raspberry Pi. Zařízení jako celek má umožňovat zobrazování informací pomocí ethernetové a bezdrátové komunikace. Vzhledem k požadavkům na univerzálnost a flexibilitu a s přihlédnutím k autorově dosavadním zkušenostem se SW vybavením byla jako nejvhodnější koncept komunikace navržena kombinace využívající HTML/PHP webové stránky, které slouží primárně pro komunikaci mezi systémem a uživatelem, a SQL databázi sloužící pro ukládání a načítání převážně zobrazovaných dat. Skrze možnosti vzdáleného přístupu přes připojenou síť je do databáze tohoto systému možné ukládat či v ní měnit data a nastavené parametry a to jak skrze webové stránky uživatelem, tak skrze přístupu od nadřazeného systému. Pro tvorbu programu starajícího se o samotné zpracovávání a přenos zobrazovacích dat do mikrokontroléru bylo využito skriptovacího jazyka Python. Znázornění výsledného systému je vidět na obr 4.1.



Obr. 4.1 Sestavení a způsob komunikace SW vybavení Rpi

Mikrokontrolér má v tomto konceptu funkci pouze tzv převodníku. To znamená že má za úkol jednoduše přijímat data a ty následně převádět na řídicí pokyny pro obvody řízení displeje, tedy vykreslit požadovaný obraz. Podle těchto požadavků byl pro mikrokontrolér navržen program, který je pro usnadnění orientace rozdělený do několika knihoven. Jako celek tak umožňuje komunikaci s Raspberry Pi a ovládání obou částí HW displeje, skrze které je zobrazování zprostředkováno, jak je možné vidět na obr.

Na závěr úvodu ještě stojí za to zmínit, že při návrhu SW vybavení bylo využíváno několika typů počítačových programů, a to jak pro přímý návrh programového vybavení, tak i pro usnadnění práce při jeho návrhu nebo analýze a následnému odstranění chyb, v kombinaci s potřebným HW vybavením. Bližší informace o nich, jako i popis jednotlivých částí podrobněji, jsou obsaženy v následujících podkapitolách.



Obr. 4.2 Znáornění řízení na straně převodníku

4.1 ATmega32

Během návrhu programové výbavy mikrokontroleru bylo v průběhu času dosaženo postupně hned několika jeho různých iterací. Výsledkem toho je nakonec program prezentovaný v této práci. Ten se sestává z několika částí, z nichž všechny jsou popsány v následujících podkapitolách.

4.1.1 Použité nástroje při tvorbě kódu

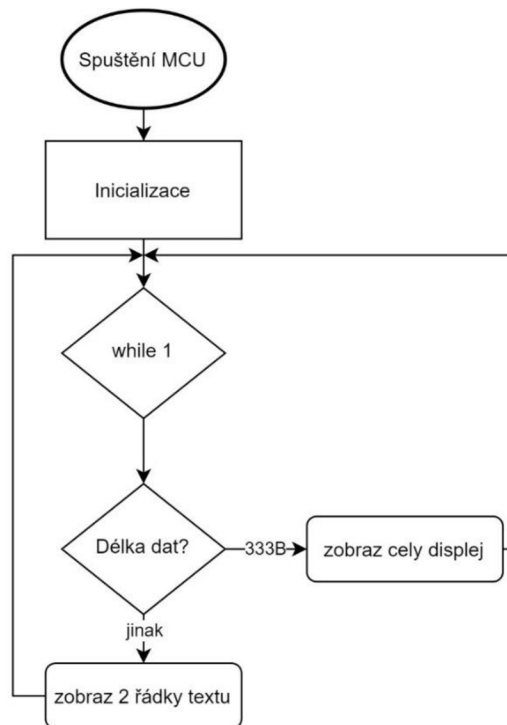
Pro mikrokontrolér bylo využito standartního vývojového prostředí pro čipy AVR s názvem AtmelStudio ve verzi 7. Kódy z něj byly poté nahrávány pomocí jednoduchého AVR převodníku označovaného jako USBasp-ISP. Jelikož tento převodník není schopen využít možnosti ladění, které prostředí nabízí, bylo během tvorby kódu a jeho ladění využito signalizačních diod na volných pinech MCU vyvedených na konektor označený jako „extension con“ a dále bylo pro ladění potíží při komunikace skrze uart využito pro mě dostupných vývojových desek platformy Arduino a jejich rozhraní Arduino IDE.

4.1.2 Tvorba kódu a jeho části

Program se sestává ze dvou hlavních knihoven. Ta první slouží pro komunikaci s nadřazeným počítačem nesoucí název „*uart*“ a ta druhá je určena pro řízení displeje a je označena „*HWdriver*“. Druhá zmíněná využívá ještě dvojici dalších knihoven. Pro překlad znaků abecedy do hodnot zobrazovaných na displeji je využívána knihovna pojmenovaná „*ASCII_table*“, v níž jsou potřebná data uložena v podobě definice a inicializace tabulky/pole parametrů. Poslední knihovna, která je mnou převzatá a částečně upravená pro moje potřeby, se skládá z maker usnadňujících zápis a čtení kódu v místech, kde je nutné ovládat jen některé z pinů mikrokontroléru, a nikoliv celý jeden port označená „*gpio*“.[30] Celý program, jako je tomu tradičně, inicializuje a spouští kód obsažený v souboru „*main.c*“. Hlavní funkce takto sestaveného programu jsou popsány v následující části. Popis každé je pro lepší pochopení doplněn o vývojový diagram či obrázek vystihující jejich hlavní funkci.

- **Hlavní smyčka (main)**

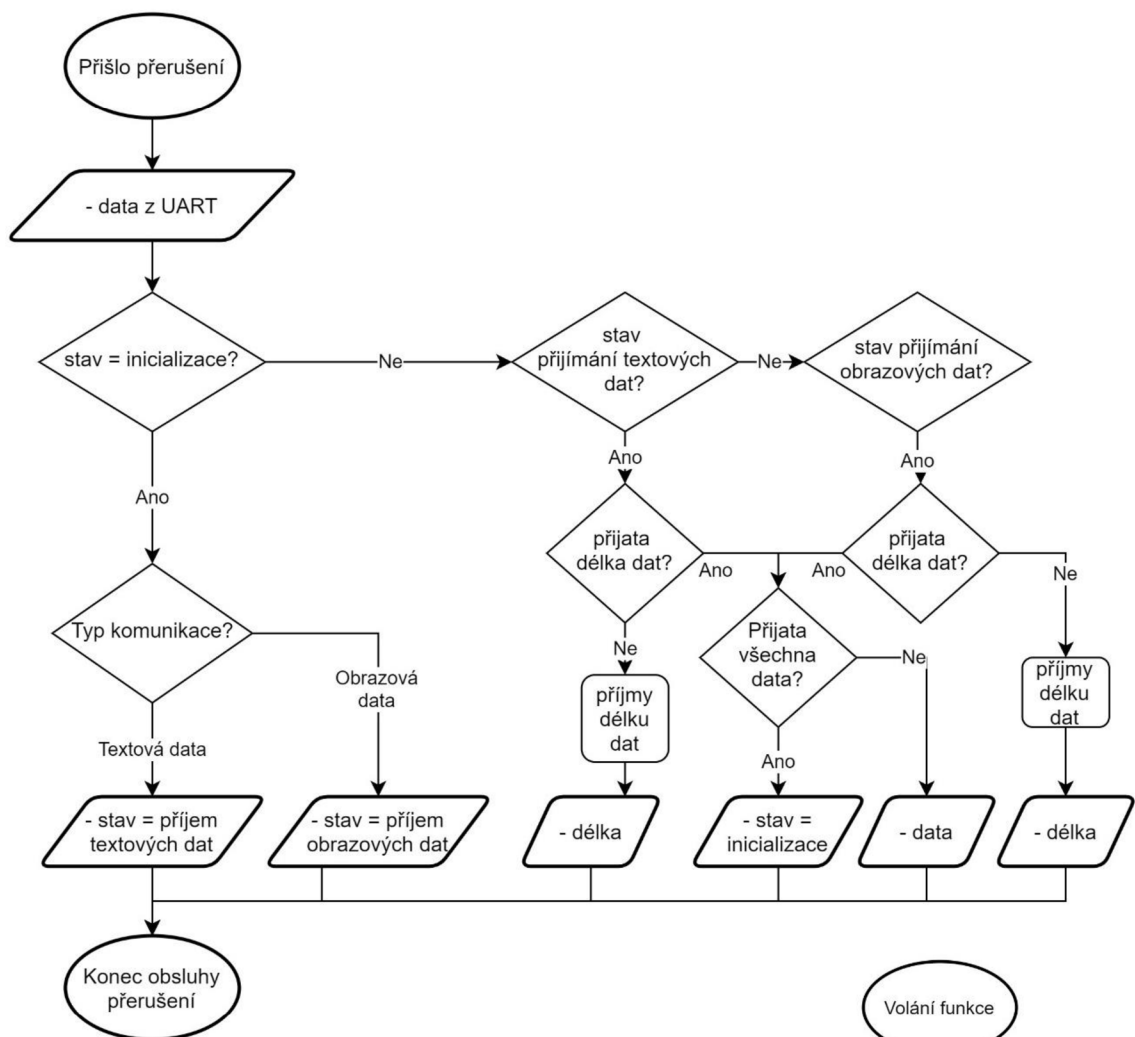
Jak je na obr. 4.3 možné vidět při spuštění mikrokontroléru, tj při přivedení napájecího napětí, nebo po uvolnění pinu RESET, dojde nejprve k inicializaci. Ta zahrnuje vytvoření globálních proměnných, nastavení vstupně-výstupních pinů a inicializaci přerušení od sériové komunikace. Poté již začíná samotná hlavní smyčka programu (while(1)), která následně spouští jednotlivé funkce v závislosti na typu požadavku přicházejícího po sériové komunikaci.



Obr. 4.3 Vývojový diagram main.c

- **Přerušení vyvolané příchozí komunikací (USART_RXC_vect)**

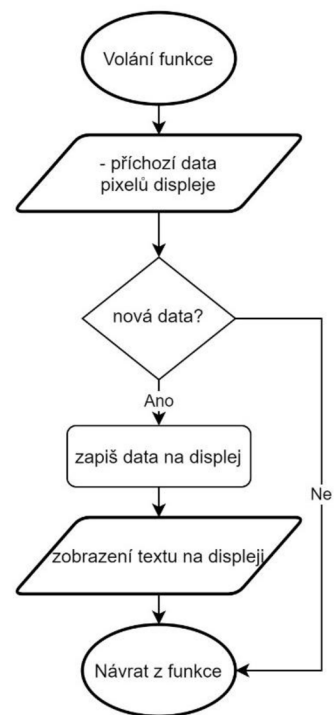
Pokud na sériové komunikaci mikrokontroléru dojde k přijetí jednoho bajtu, tak ten se uloží do registru a následně se vyvolá přerušení. Díky tomu se nestane, že by mikrokontroler daný bajt nestihl obsloužit. Příjem zprávy je rozdělen do tří částí. Nejprve se přečte 1. bajt, díky němuž se nastaví typ komunikace. Následuje 2. bajt informující o délce příchozích dat a poté jsou již přijímány samotná data až do posledního, jak je možné vidět na obr. 4.5. Jelikož jde o obsluhu přerušení jsou veškerá příchozí data ukládána přímo do globálních proměnných.



Obr. 4.5 Vývojový diagram příchozí komunikace

- **Funkce pro zobrazení celého obrazu (zobraz_cely_disp)**

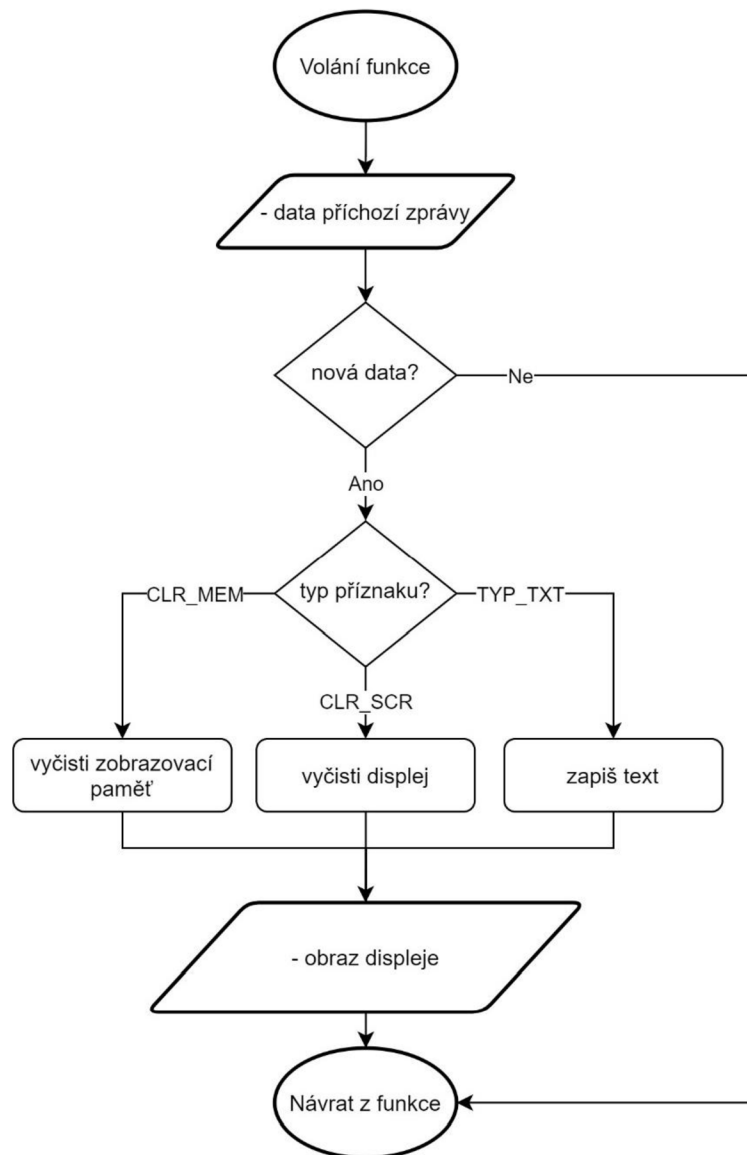
Pro případ, že je v příchozích datech již celý obraz zakódován, tedy byl sestaven v nadřazeném mini počítači RPi, spočívá vykreslení obrazu pouze v jednoduchém dekódování a zápisu bodů na displej, jak je vidět na obrázku 4.4. K samotnému zobrazení je využívána funkce HW_zapis.



Obr. 4.4 Vývojový diagram zobraz_cely_disp

- **Funkce pro příjem 2 řádků textu (zobraz_dva_radky)**

Pro využití již navržené funkce HW_vytvor_disp a k možnosti přímého zobrazení textu na displej slouží tato funkce. Pomocí ní lze vymazat vnitřní paměť MCU pro ukládání obrazových dat, jednoduchým povelům vymazat celý displej a hlavně zapsat 2 řádky textu na displej. Ty jsou omezeny maximální délkou 2x26 znaků a odděleny od sebe ASCII znakem 0xd (návrat vozíku).



Obr. 4.6 Vývojový diagram zobraz_dva_radky

- **Funkce pro sestavení obrazu z textu (HW_vytvor_disp)**

V prvotní fázi návrhu bylo zamýšleno, že v případě požadavku na zobrazení jednoduchého textu bude možno zobrazení pro tento text složit až v samotném mikrokontroléru, což by systému poskytovalo možnost jeho propojení s různými

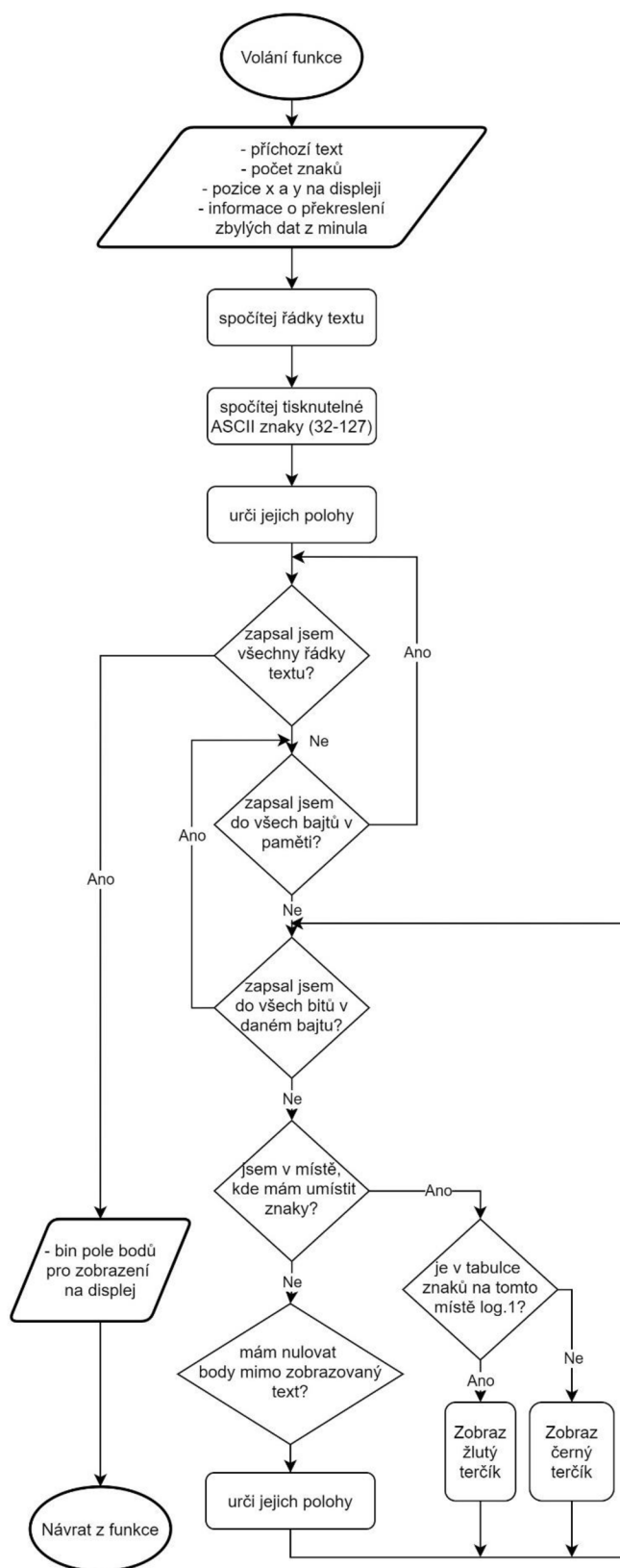
řídícími počítači. Proto vznikla funkce, která pro zobrazení textu vyžaduje od nadřazeného počítače pouze text a jeho pozici, z nichž následně s pomocí vlastní tabulky textu nazvané `ASCII_table` připraví obrazová data pro funkci `HW_zapis`. Od toho bylo v pozdější fázi upuštěno pro svou nepraktičnost a do budoucna nedostatečnou paměť na případné další fonty. Funkce však jak již bylo uvedeno zůstala zachována a je pro její demonstraci využívána pro příjem 2 řádků textu.

Funkce tedy nakonec na základě příchozích znaků textu jen ukládá bity, které jí poskytuje font uložený v tabulce. Zkrátka pomocí binární hodnoty aktuálního znaku najde v tabulce pro daný znak 5 sloupců dat (jedná se o font 5x7) a ty vhodně zapíše do paměti zobrazení nesoucí název pole `_pixelu`. Mezi znaky dělá mezeru jeden bod širokou.

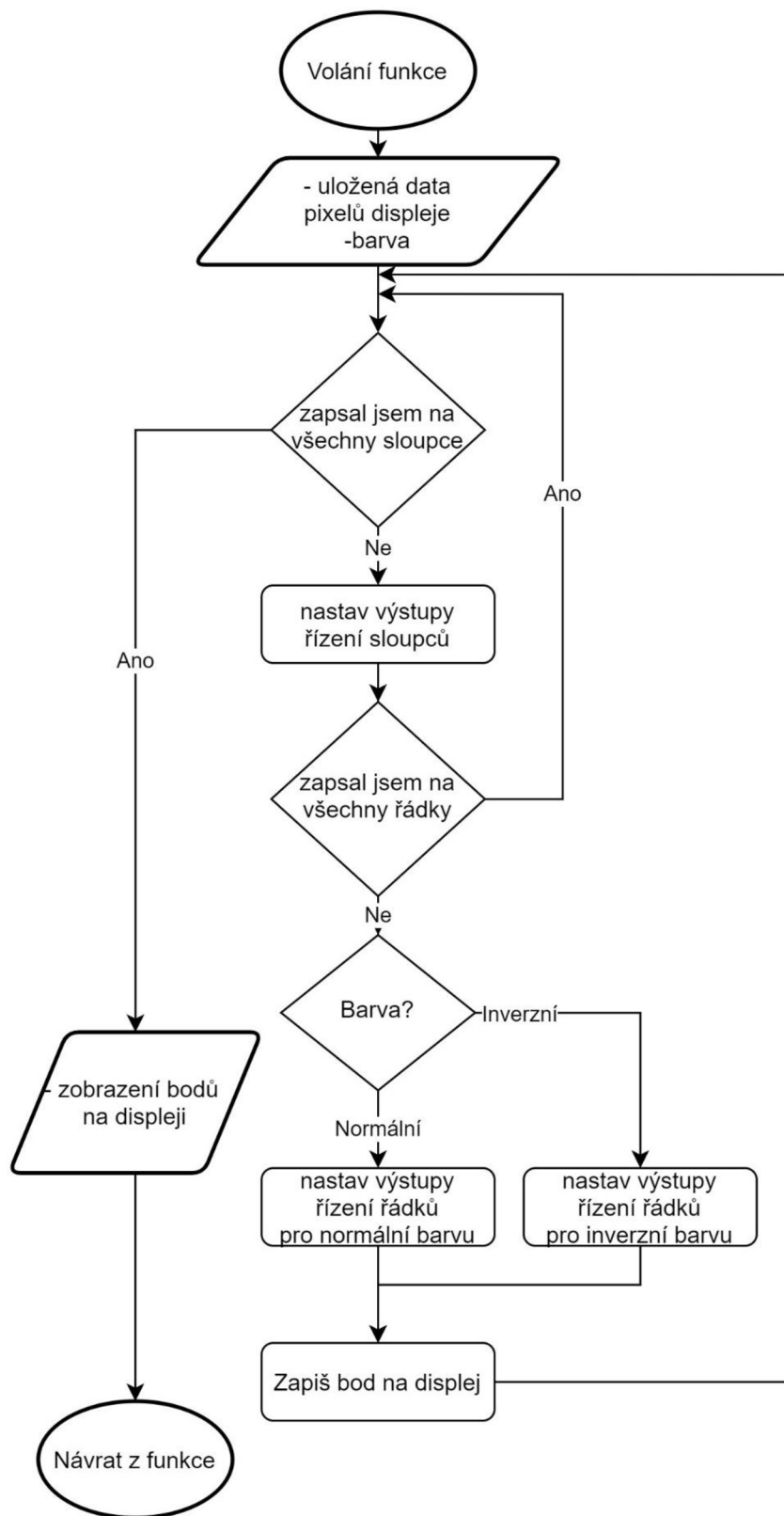
Podrobněji je tato funkce popsána ve vývojovém diagramu, který je možné vidět a obr. 4.7 na další straně.

- **Funkce pro zobrazení celé paměti dat na displej (`HW_zapis`)**

Tato funkce se stará o samotný zápis do výstupních registrů mikrokontroleru. Je tak spouštěna pokaždé, kdy je potřeba změnit zobrazení displeje. Pro zjednodušení zápisu je v kódu využita knihovna `gpio`. Funkce prochází postupně všechny body displeje a testuje, zda je na dané pozici v paměti binární hodnota 0 či 1. V případě že se zde nachází hodnota 0 daný terčík na displeji se ven otočí černou barvou, v opačném případě barvou žlutou. Pro případ že je funkci dán požadavek pro zobrazení barvy negativu (tzn. Černá místa na žlutém pozadí) provede se předchozí krok opačně. Zápis samotný je časově omezen pomocí čítače/časovače označeného `Timer2`. Ten je nastaven v inicializaci a při připojení požadovaného napětí na požadovaný bod displeje nastaven na požadovanou dobu. Nastavení této doby je popsáno v části 4.1.3. Při uplynutí této doby se vyvolá přerušení, které má za následek okamžité odpojení řízeného napětí a přesun na další zapisovaný bod. Takto zapíše na všech 2660 zobrazovaných bodů. Vývojový diagram pro lepší pochopení je na obr. 4.8.



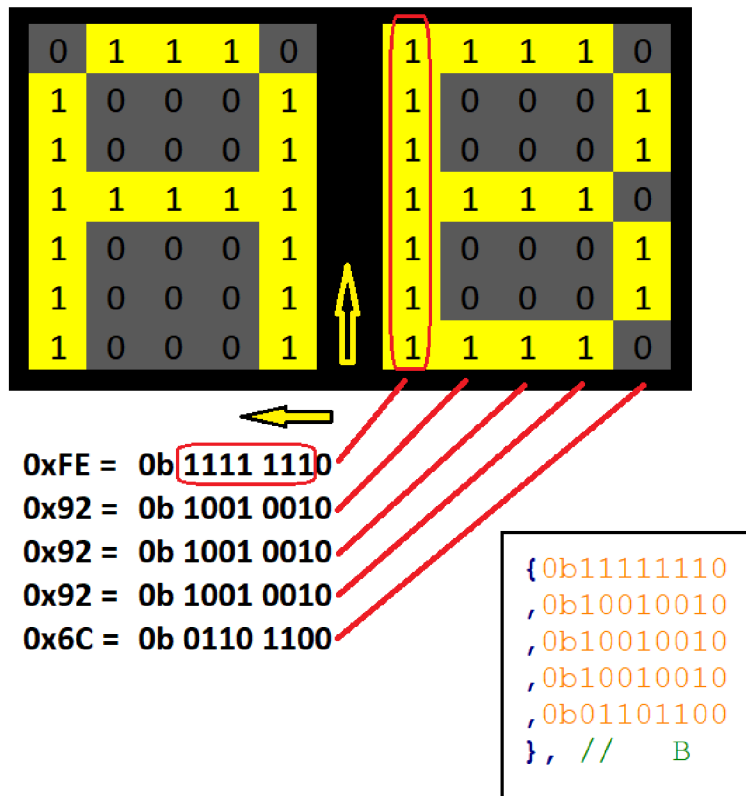
Obr. 4.7 Vývojový diagram HW_vytvor_disp



Obr. 4.8 Vývojový diagram HW_zapis

- **Struktura dat uložených znaků (ASCII_table)**

Pro účely vytváření obrazu z textu byl do mikrokontroléru uložena tato tabulka v podobě pole. Způsob reprezentace zobrazovaných bodů do uložených dat v proměnné ASCII_table byl pro zjednodušení jejího převodu do paměti pole_pixelu strukturován podobně. Každý pixel má tedy v uloženém poli vždy 5 bajtů, tedy 8x5 bitů. Jelikož pro účely zobrazování postačí 7x5 bitů, je v každém bajtu nultý bit roven 0 a při převodu je zanedbáván. Nejlépe tento zápis vidět na obr 4.9.



Obr. 4.9 Způsob uložení fontu v MCU

- **Knihovna gpio.h [30]**

gpio.h využívá ke své funkci textová makra, která jsou při kompilaci programu zpracovávána na začátku pomocí preprocesoru. Ten se stará o nahrazení identifikačního řetězce za obsah daného makra. Navíc v případě že je makro napsáno jako funkce, a tedy obsahuje vstupní parametry, jsou názvy označující tyto parametry vkládány do míst výskytu v daném makru. Díky tomu jsou všechny potřebné části kódu uvedeny pouze v hlavičkovém souboru.

4.1.3 Časovač Timer2

Atmega32 obsahuje hned 3 časovače označené „Timer/Counter“ a čísla 0, 1 a 2. Pro potřeby časování doby průchodu proudu cívkou terčíku snadno postačí pouze 8bitový časovač s využitím děličky frekvence označované anglicky jako „prescaler“. Proto byl

vybrán právě časovač Timer2.

Z několika možností nastavení jeho funkce bylo vybráno jednoduché přetečení s přednastavením jeho počáteční hodnoty. Pro výpočet doby časování lze využít jednoduchou rovnici sestavenou pomocí katalogového listu MCU. [23] Výpočet je potom následovný:

$$t_{ON} = \frac{2 \cdot N_{del} \cdot (TCV_{max} - TCV_{set})}{f_{clk}} \quad 3$$

$$t_{ON} = \frac{2 \cdot 64 \cdot (256 - 220)}{14\,745\,600} = 312,5 \mu s$$

, kde:

N_{del}	je hodnota děličky	(-)
TCV_{max}	je maximální hodnota registru čítače	(-)
TCV_{set}	je přednastavená hodnota registru čítače	(-)
F_{clk}	je frekvence připojeného krystalu (frekvence MCU) (Hz)	
t_{ON}	je čas potřebný pro vyvolání přerušení	(s)

Vzhledem k omezeným možnostem měření byly hodnoty přednastavení časovače a děličky vybrány jen hrubě, a to jednoduchou metodou postupného prodlužování doby působení napětí, tak aby byl výsledný obraz čitelný při jednom průchodu zápisu a hodnotě napětí 24 V.

4.1.4 Struktura dat uložených zobrazovaných bodů proměnné pole_pixelu

Protože původním záměrem bylo umožňovat řídicímu počítači zasílat různé požadavky na změny zobrazení bylo potřeba v mikrokontroleru umět uložit data pro aktuální stav všech bodů displeje. Z důvodů malé paměti mikrokontroléru byla od počátku snaha o využití jednoduché paměťové komprese zobrazovaných bodů. Místo využití nejmenšího typu proměnné pro každý zobrazovaný bod, byl každý bod ukládán do jednoho bitu. Toho lze docílit při zápisu do proměnné pomocí bitových operací.

Pro zápis binární hodnoty 1 do náhodného bitu „n“ v proměnné velikosti jeden bajt stačí zapisovaný bit tzv posunout na pozici přepisovaného bitu, čímž dočasně vznikne proměnná potřebné velikost, v nichž jsou všechny ostatní bity nulové, a poté do přepisované proměnné uložit bitový součet (OR) původní a dočasné proměnné. Tento složitý výklad lze pro zasvěceného čtenáře popsat jednoduchým zápisem v jazyce C:

$$y | = (1 \ll n) ;$$

Pro zápis binární hodnoty 0 lze postupovat velice podobně s tím rozdílem, že dočasně vzniklou proměnnou ještě znegujeme čímž v požadovaném místě vznikne bin hodnota 0 a na ostatních pozicích samé jedničky. Následně použijeme logický součin

(AND) a tím binární hodnotu v daném místě vynulujeme. Zápís takového kódu pak vypadá následovně:

$$y \& = \sim(1 \ll n);$$

Oba způsoby výpočtu jsou pro lepší názornost zobrazeny na obr. 4.10.

0xAA = 0b 1010 1010

0xAB = 0b 1010 1011

Zápis binární hodnoty 1:

Zápis binární hodnoty 0:

pořadí: 7654 3210

pořadí: 7654 3210

0b 0000 0001 << 2 = 0b 0000 0100

~(0b 0000 00001 << 0) = 0b 1111 1110

or

and

0b 1010 1010

0b 1010 1011

0b 1010 1110 = 0xAE

0b 1010 1010 = 0xAA

Obr. 4.10 Výpočet užívaný při binárním zápisu do proměnné

Díky těmto operacím je možné pro uložení dat obrazovky ušetřit až 7/8 paměti. Pro případ displeje BS110 je tak místo 2600 bajtů, potřeba pouze 333 bajtů. Jednotlivé body jsou pak uloženy v jednom poli jednobitových hodnot od terčiku S1R1 po terčik S1R8 v prvním bajtu, od S1R9 po S1R16 ve druhém atd., přičemž při dosažení posledního řádku se plynule přejde beze změn na řádek nový. Nejlépe je možné tento způsob pochopit pomocí grafického znázornění na obr. 4.11. Každý čtvereček na obrázku reprezentuje jeden bit v paměti uložený na pozici dané tvarem [číslo bajtu , číslo bitu].

	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10
r19	2,2	4,5	7,0							
r18	2,1	4,4	6,7							
r17	2,0	4,3	6,6							
r16	1,7	4,2	6,5							
r15	1,6	4,1	6,4							
r14	1,5	4,0	6,3							
r13	1,4	3,7	6,2							
r12	1,3	3,6	6,1							
r11	1,2	3,5	6,0							
r10	1,1	3,4	5,7							
r9	1,0	3,3	5,6							
r8	0,7	3,2	5,5							
r7	0,6	3,1	5,4							
r6	0,5	3,0	5,3							
r5	0,4	2,7	5,2							
r4	0,3	2,6	5,1							
r3	0,2	2,5	5,0							
r2	0,1	2,4	4,7							
r1	0,0	2,3	4,6							

Obr. 4.11 Způsob uložení informace o obrazu displeje

4.1.5 Sériová komunikace

Pro odesílání požadavků a dat z RPi do MCU je potřeba ještě nastavit parametry asynchronní komunikace označované jako UART, a to na obou stranách. Vzhledem k malému objemu přenášených dat není nutné dosahovat velkých přenosových rychlostí. Z tohoto důvodu jsou v práci použity následující parametry. Baudrate 9600 bits/s, délka dat paketu je 8 bitů a 1 stop bit. Parity nakonec v rámci testování prototypu nebylo využito, tudíž bez paritního bitu. O funkci inicializace této komunikace se stará funkce „UART_Init“.

Všechny vytvořené zdrojové kódy jsou dostupné v příloze G v projektu s názvem „Diplomka_ATmega32_V5 (zV3.9)“

4.2 Raspberry Pi 3 B+

4.2.1 Použité nástroje při tvorbě kódu

Minipočítač RPi, na rozdíl od mikrokontroléru, nabízí mnohem větší možnosti nejen co se týče výpočetního výkonu, či využití periférií, ale i různých typů programovacích jazyků. I proto bylo při návrhu SW vybavení pro RPi bylo využito hned několika různých nástrojů. Nástroj *Etcher* pro zápis obrazu operačního systému „*Raspbian Lite*“ na SD kartu sloužící u RPi jako hlavní paměť. Pro vzdálenou komunikaci mezi RPi a počítačem bylo využito multi tabulkový SSH klient *SmarTTY*. Z důvodů využití webových stránek byl na RPi nainstalován balíček tzv. *LAMP* obsahující zejména *Apache* server, *PHP*, odnož *MySQL* databáze s názvem MariaDB. Pro návrh webových stránek v PHP bylo využito textového editoru zdrojových kódů *Atom* spolu s rozšířením „*remote-FTP*“, které umožňuje úpravu zdrojového kódu v Raspberry Pi přímo ze vzdáleného počítače. K tomu navíc ještě multiplatformní softwarový balíček *XAMPP* pro ladění webových stránek přímo v počítači. V poslední řadě je v RPi užito nástroje *phpMyAdmin*.

4.2.2 Příprava RPi pomocí dostupného SW vybavení

Ačkoliv je Raspberry Pi po HW stránce velmi dobře vybaven neobsahuje po zakoupení žádný úložný prostor, a tudíž neobsahuje ani žádný předinstalovaný OS či jiný SW. Jako paměť slouží SD paměťová karta jednoduše zasunutá do připraveného slotu. Tu je tedy nejprve potřeba vybrat podle požadavků a připravit. [31]

Pro instalaci OS je potřeba nejprve stáhnout jeho obraz. Ten je možné stáhnout z oficiálních stránek a nainstalovat pomocí dostupných programů jako např. Etcher.

Jelikož RPi nebude užíváno jako osobní počítač s klávesnicí a displejem, nýbrž vzdálený server, lze se k němu pro tuto možnost připojit z jiného počítače pomocí SSH. Nejprve přes Ethernet, kdy je možné nastavit bezdrátové připojení a díky tomu je k němu možné následně přistupovat i přes Wifi. Ke komunikaci je možné využít třeba SmarTTY klienta, který práci usnadňuje.

Pro možnosti běhu PHP webu, MySQL databáze a Python skriptů je na RPi dále potřeba nainstalovat tyto funkce. K tomu u RPi slouží balíček LAMP, jehož instalace zahrnuje několik kroků popsaných například ve zdroji [32].

V případě řízení RPi vzdáleně, tedy jako sever, je buď nutné upravit vlastnictví a oprávnění v daných složkách, nebo stačí před každým požadavkem uvést klíčové slovo „sudo“, které opravňuje uživatele jako tzv. super uživatele (root). Pro umožnění přístupu ostatních uživatelů jako např. www-data (Apache) do jednotlivých složek je potřeba již oprávnění a vlastnictví upravit. Přehled základních příkazů jako jsou ty pro úpravu vlastnictví (chown) a pro úpravu oprávnění (chmod) lze nalézt ve zdroji [33].

Pro UART komunikaci s mikrokontrolerem ATmega32 slouží piny 8 a 9, tedy GPIO14 a GPIO15. Ty je však možné u Raspberry Pi přiřadit k jedné ze dvou UART komunikací. Jedna z nich nese název „mini UART“ nebo také „UART1“ a je u verze RPi3B+ označena jako primární. Ta druhá se nazývá „PL011“ nebo také „UART0“, a i když je označena jako sekundární, jedná se ve skutečnosti o schopnější verzi, která je na rozdíl od té předchozí ve výchozím stavu povolena a využívána pro Bluetooth komunikaci. Bluetooth komunikace však není v této práci zapotřebí, jelikož bezdrátové spojení bude realizováno skrze Wi-Fi, a proto ji je možno jednoduše zakázat, či přesunout na mini UART, a uvolnit tak PL011 pro komunikaci s mikrokontrolerem ATmega32. Vypnutí Bluetooth se docílí jednoduchým připsáním řádku kódu „dtoverlay=pi3-disable-bt“ do souboru „/boot/config.txt“.[34] Ten zajistí nejen deaktivaci Bluetooth ale i přiřazení primárního označení pro PL011, díky čemuž k němu lze přistupovat jako „/dev/serial0“. V poslední řadě se v tomto souboru ještě nastaví parametry sériové komunikace, které jsou popsány v podkapitole 4.1.5.

4.2.3 Struktura internetových stránek

Pro komunikace s uživatelem byl vybrán způsob využívající webových stránek běžících na počítači RPi. To se skládá ze statické kostry tvořené HTML stránkami, které jsou propojeny hypertextovými odkazy. Vzhled této kostry dále upravují tzv. kaskádové styly označované jako CSS. Aby mohli být takto sestavené stránky dynamické, tedy mohli provádět změny na straně serveru, je tato dvojice doplněna o tzv. hypertextový preprocesor označovaný PHP. Celé takto sestavené stránky jsou potom dostupné skrze nainstalovaný server Apache.

4.2.4 Popis částí internetových stránek

Výsledná struktura webového serveru se sestává z následujících souborů a složek. Základem jsou hlavní PHP soubory obsahující HTML kostru, a k nim dále pomocné soubory PHP vykonávající pro jednotlivé stránky různé pomocné funkce. Ty jsou pojmenovány ve tvaru „*nazevStránky.typFunkce.inc.php*“. Přístup k nim je vždy podmíněn pomocí PHP, a proto k nim nelze jednoduše přistupovat jednoduše skrze jejich

adresy. Díky tomu je znemožněno se po stránkách náhodně procházet. Hlavními soubory webových stránek jsou:

- ***header.php***

Je soubor pro zobrazení hlavničky. Na každé stránce je tento soubor vložen a díky tomu je vzhled načítaných stránek konzistentní a lze jej měnit jen na jednom místě.

- ***footer.php***

Stejně tak je tomu i v případě tohoto souboru pro zápatí stránek. V případě naimplementovaného vzhledu nese pouze informaci o roku vypracování.

- ***navigation.php***

I v tomto případě se jedná o soubor vkládaný do všech stránek s výjimkou přihlašovací stránky, protože neautorizovaná osoba nemusí vědět nic o struktuře tohoto webu. Díky tomuto souboru je na každé stránce zobrazen v pravém sloupci navigační panel. Ten se skládá z odkazů na 3 základní stránky které jsou pojmenovány jako Úvod, Projekty a Přímé ovládání. Pod nimi se nachází odkaz pro přihlášení do správy dat uložených v databázi a odkaz pojmenován Odhlásit, který se stará o odhlášení uživatele z internetových stránek.

- ***Styles.css***

Na první pohled je jasné, že se tento soubor liší od všech ostatních a to příponou .css. Jedná se totiž o soubor pro vytváření kaskádních stylů webové stránky. Díky němu lze v souborech PHP upravovat vzhled HTML části, a to hromadně pro celý web. Jsou v něm předdefinovány grafické vzhledy jednotlivých částí webových stránek. V rámci tohoto webu je v něm nastaven vzhled hlavičky, navigačního panelu, pozadí stránky, vzhledu tabulek a některých dalších částí jako jsou paragrafy aj.

- ***index.php***

Jedná se o soubor, který se zobrazuje vždy jako první a jedná se tak o vstupní bránu do webových stránek. V těchto stránkách však slouží pouze pro přesměrování na stránku pro přihlášení uživatele.

- ***login.php***

Tato stránka slouží pro přihlášení přiděleného uživatele. Tento krok je podmínkou pro umožnění přístupu do stránek určených k ovládání displeje. Tato funkce je užitečná při provozování zařízení v síti s více uživateli, z nichž jen někteří mají být oprávněni displej ovládat.



Ovládání BUSE110

Vítejte

Tato stránka je určena k řízení displeje BUSE110 prostřednictvím webového prohlížeče.
Pro možnosti řízení displeje je nutné se přihlásit pomocí přidělených přihlašovacích údajů

Přihlášení

2020-2021

Obr. 4.12 Ukázka stránky login.php

V případě, že uživatel není přihlášen nemá přístup k žádné jiné ze stránek a je vždy přesměrován na tuto výchozí stranu. Toho je docíleno i v případě, že se pokusí si pomocí zadáním přímých adres ostatních stránek. Pro tuto funkci je při načítání každé z přímo přístupných stránek volán soubor „*login.check.inc.php*“, který jen jednoduše ověří, zda je uživatel přihlášen. Pokud ano, tak příslušná stránka zobrazí, pokud ne, tak jej vrátí na začátek webu.

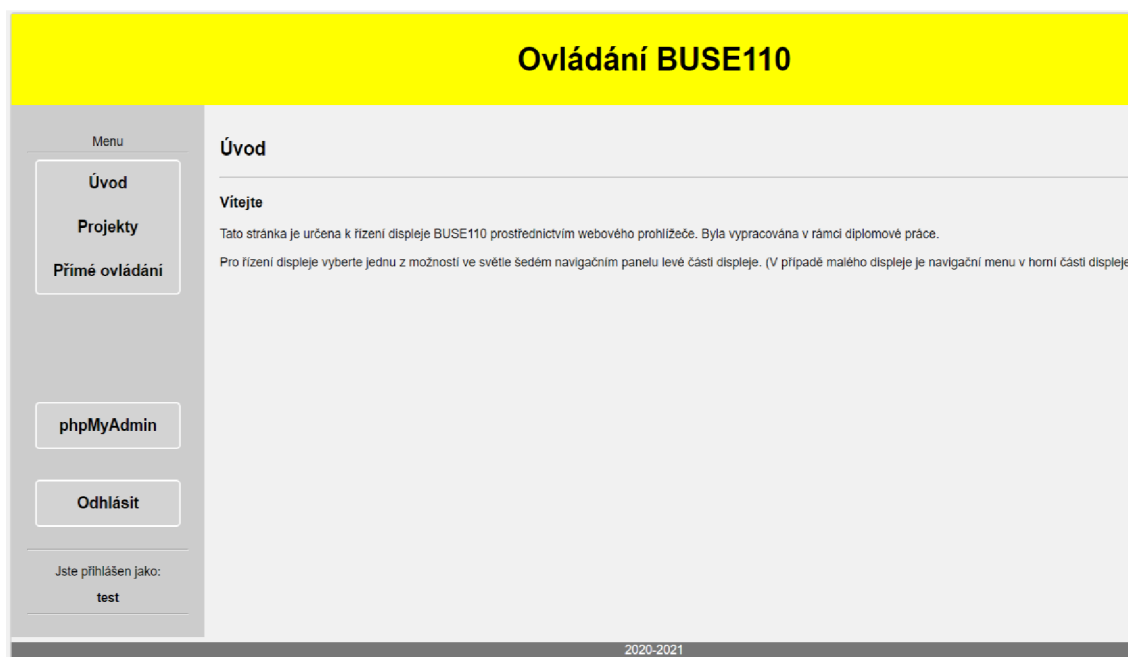
Pro samotné přihlášení je volán soubor „*login.login.inc.php*“ využívající funkce ověření přihlašovacích údajů a následného přihlášení, které obsahuje soubor „*login.functions.inc.php*“. Ten umí zjistit, zda jsou přihlašovací údaje vyplněny a odpovídají záznamům uloženým v databázi pojmenované „*dbloginsys*“ do které přistupuje pomocí souboru „*login.dbh.inc.php*“.

V případě, že dojde ke chtěnému odhlášení uživatele prostřednictvím tlačítka „Odhlásit“ dojde k volání souboru „*logout.login.inc.php*“, který záznamy o přihlášení jednoduše zruší pomocí funkce PHP označené „*session_destroy()*“.

Pro zvýšení bezpečnosti by v případě potřeby bylo vhodné doplnit přístupové údaje o tzv hešování (z anglického Hash) hesel. To by zabránilo posílání přihlašovacích údajů přímo a zasílalo místo nich pouze jejich tzv otisky. Tento krok byl pro zjednodušení tvorby přístupových údajů opomenut.

- *main.php*

Po úspěšném přihlášení do systému se načte hlavní stránka nazvaná „Úvod“ uložená v souboru main.php. Ta obsahuje pouze uvítání a krátký úvod pro uživatele.

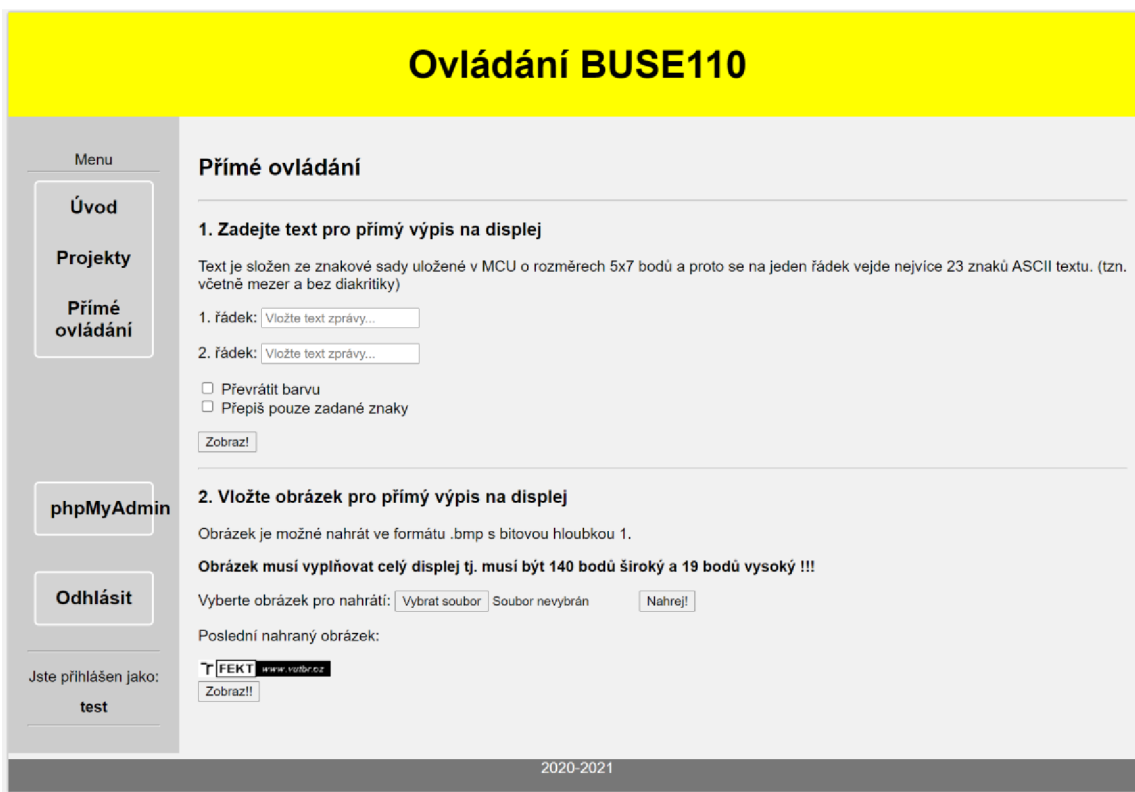


Obr. 4.13 Ukázka stránky main.php

- ***pages/prime.php***

Toto je první ze dvou stránek určených pro ovládání dat zobrazovaných na displeji. Jde také o první ze dvou způsobů řízení displeje. Jak název sám napovídá jedná se o přímé řízení zobrazení, čímž je myšleno jednorázové zobrazení zadaných hodnot. Stránka je opticky rozdělena na dvě části samostatné části.

První část je výsledkem prvních pokusů o řízení displeje. Umožňuje uživateli využívat displeje k přímému zobrazení zprávy pomocí fontu s rozměry 5x7 bodů, který jak již bylo zmíněno je definovaný v knihovně „ASCII_table“ a uložený v MCU převodníku. Z rozměrů fontu je pak jasné, že na displej se nad sebe vlezou právě 2 řádky, z nichž každý může čítat až 23 zobrazitelných ASCII znaků. Toto jsou omezení dána stávající realizací. Jelikož MCU dále umí zobrazovat data i v převrácených barvách (tzn místo žlutého textu na černém pozadí lze zobrazovat i černý text na žlutém pozadí) a vytvářet nově vzniklé zobrazení pomocí změn aplikovaných pouze v místech ve kterých má docházet ke změnám, jsou tyto možnosti na výběr i tyto možnosti. O zasílání vložených dat se stará soubor „prime.inc.php“.



Obr. 4.14 Ukázka stránky prime.php

Druhá z část přímého zápisu na displej je určena pro zobrazování obrazových dat. Pro tuto funkci slouží možnost odeslat a uložit na server vždy jeden bitmapový obrázek s příponou „.bmp“ s nejnižším pixelovým formátem označovaným jako 1bpp, tedy 1 bit dat na pixel. Obrázek je tedy pouze dvoubarevný a skládá se z bodů bílé a černé barvy. Dále je nutné, aby splňoval stejný počet bodů na výšku a šířku, jako je tomu u ovládaného displeje a tím byly všechny body displeje jednoznačně určeny. Proto lze nahrávat obrázky pouze o velikosti 140x19 bodů. O nahrávání a kontrolu typu a velikosti nahrávaného obrázku se stará soubor „prime.upload.inc.php“. V případě nahrání obrázku se správnými parametry je obrázek uložen na web do složky „img“ s názvem

img_prime_ovladani.bmp. Tento obrázek je poté možno vidět ve spodní části obrazovky a pod ním přidané tlačítko pro jeho zobrazení na displej po jehož stisknutí se pomocí souboru „prime.show.inc.php“ spustí jednorázový běh skriptu s názvem „bmp.py“.

- **pages/projekty.php**

Tato stránka je určena pro druhý způsob řízení displeje. Jde o řízení mnou označené jako Projekty a zahrnuje komunikaci s místní databází, která poskytuje data pro automatické zobrazování informací v předdefinovaných formátech. Samotné procesy získávání dat z databáze, jejich úpravu a odesílání do MCU jsou realizovány pomocí skriptu nesoucího název „app.py“. Zobrazená stránka se skládá opět z několika část.

V horní části je nejprve zobrazen název posledně vybraného projektu. Pod ním je zobrazen stav aktuálního projektu, který je získáván pomocí souborů uložených ve složce status. V případě chyby či nečekaného ukončení běhu systému je možné se dostat do stavu kdy zůstal od posledně soubor „script_running“ a při pokusu o zpuštění nového projektu

id	nazev	typ	idDat	barva	poziceX	poziceY
1	demo_nadpis_h	0	1	0	1	1
2	demo_rok_s	0	2	1	1	11
3	demo_web	0	3	0	60	11
4	demo/VUT2	0	4	0	2	1
5	demo/fekt1	0	5	1	1	11
6	demo_nadpis_s	0	1	0	1	11
7	demo_rok_h	0	2	1	1	1
8	demo_logo_vut	1	1	1	0	0
9	demo_logo_fekt	1	2	0	19	0
10	demo/VUT1	0	4	0	2	10
11	demo/fekt2	0	5	1	1	1
12	demo_ramecek	1	3	0	0	0
13	demo_obrazek_PM00	1	4	0	0	0
14	demo_obrazek_PM01	1	5	0	0	0
15	demo_obrazek_PM02	1	6	0	0	0
16	demo_obrazek_PM03	1	7	0	0	0
17	demo_obrazek_PM04	1	8	0	0	0

Obr. 4.15 Ukázka stránky projekty.php

či ukončení stávajícího vznikne požadavek „stop_script“, který však nebude nikdy obslužen vzhledem tomu že skript app.py neběží. Pro tento a podobné případy bylo

v případě detekce chyby přidáno pod ukazatel stavu ještě tlačítko reset které spouští soubor „*projekty_reset.php*“, který se stará o inicializaci stavů skriptu *app.py*.

V další části je umístěno tlačítko pro ukončení zobrazování vybraného projektu. Tuto funkci obstarává soubor „*projekty_stop.php*“. (Pozn. Podtržítka namísto tečky označuje, že se jedná o stránku ovládající hlavní skript.)

Poté následuje část se seznamem názvů projektů uložených v databázi, z nichž vedle každého z nich se nachází tlačítko pro jeho okamžité spuštění. To je zprostředkováno pomocí souboru „*projekty_start.php*“. Ten obdrží od stránky „*projekty.php*“ informaci o identifikátoru vybraného projektu a s využitím databáze *dbprojekty* nastaví nové parametry do databáze *dbnastaveni* sloužící pro řízení skriptu *app.py*.

V poslední části jsou pak zobrazeny textová a obrazová data tabulek uložených v databázi *dbprojekty*. Ta slouží pro částečné usnadnění přehledu v již uložených datech a to především pro fázi testování. Tento způsob je totiž vhodný jen pro velmi malé množství dat. Při větším množství dat začíná být nepřehledný a spíše zdržuje, a tak by bylo dobré jej do budoucna prezentovat pomocí vhodného uživatelského rozhraní.

- ***status***

Jedná se o složku, která obsahuje soubory sloužící k identifikaci stavů programu „*app.py*“. Podle daného stavu obsahuje soubory jako jsou „*script_stopped*“, popř. „*script_running*“, které informují o aktuálním stavu, a „*stop_script*“, který značí požadavek na ukončení běhu programu „*app.py*“.

- ***includes***

Jak již bylo zmíněno výše, jedná se o složku obsahující PHP soubory, které jsou využívány v rámci funkcí jednotlivých internetových stránek. Jedná se především o úkony způsobené požadavky v jednotlivých HTML stránkách jako např. požadavek po stisku tlačítka. Pochází jmenovitě především z *login.php*, *prime.php* a *projekty.php* ale jsou využívány i ve zbylých částech stránek.

- ***img***

Tato složka je určena pro ukládání obrázků užívaných v tzv. přímém ovládní displeje. V současném stavu obsahuje vždy pouze jeden, a to poslední nahraný, obrázek nesoucí název „*img_prime_ovladani.bmp*“.

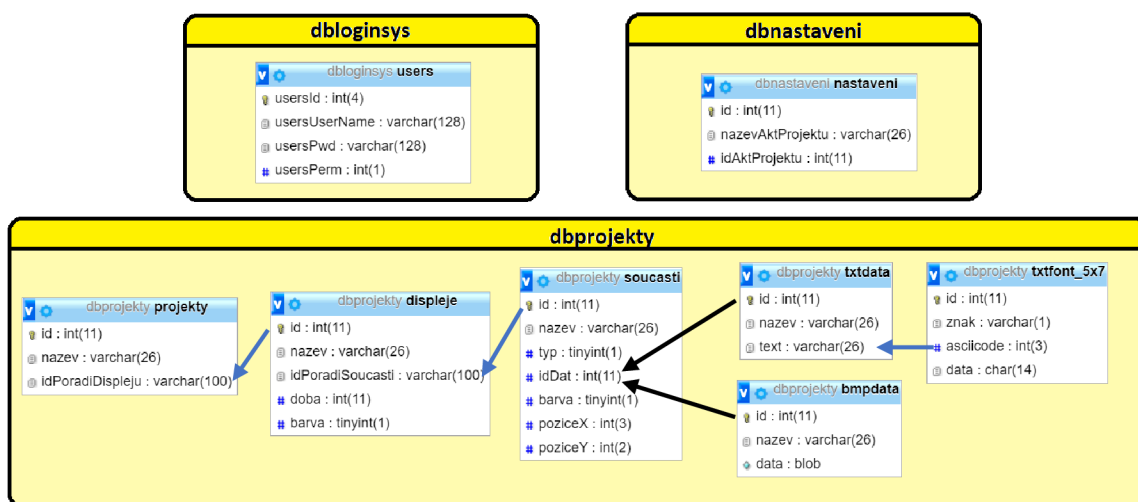
Závěrem

4.2.5 Databáze

Pro usnadnění práce s databází bylo využito nástroje pro jednoduchou správu nazývaného *phpMyAdmin*. Tu lze do RPi jednoduše nainstalovat například podle zdroje [35]. Díky ní byly vytvořeny databáze a v nich tabulky pro všechna potřebná data. Jejich struktura je vidět na obr. 4.16. V obrázku jsou zobrazeny jednotlivé databáze včetně názvů tabulek a jejich sloupců. Díky rozdělení tabulek do několika databází je možné různým uživatelům umožnit různá oprávnění. Například internetové stránky do databáze přistupují skrze uživatele „*webuser*“ a mají oprávnění číst (*SELECT*) *dbloginsys* pro přihlašování a

dbprojekty pro zobrazování struktury dat na stránce projekty.php. Pro databázi dbnastavení mají navíc oprávnění tabulku aktualizovat (UPDATE).

Kromě zobrazení částí databáze je v obrázku pro databázi dbprojekty znázorněna hierarchie a provázání prvků, využívané pro zobrazování přednastavených projektů. Zprava jsou tabulky s nejjednoduššími prvky jako jsou znaky (včetně jejich fontu - tabulky „txtfont_5x7“) a obrázky. Nad nimi (resp. vlevo od nich) leží tzv. součásti, z nichž každá využívá pouze jeden text, či obrázek a vytváří z nich součást. Součásti již mají definovanou polohu a barvu. Z nich se pak skládají jednotlivé obrazovky, která mají navíc přidělenou dobu zobrazení na displeji a opět mohou měnit barvu všech prvků zároveň. Jednotlivé obrazovky se pak shlukují do projektů, které definují posloupnost zobrazení obrazovek a mohou být spouštěny skrze web. Barva šipek v obrázku značí pro



Obr. 4.16 Znázornění tabulek databází a jejich vazby

modré šipky – přiřazení vždy jen jednoho nižšího prvku, pro černé šipky – výběr celé posloupnosti prvků. Při návrhu projektu je tak důležité dodržet tyto zákonitosti. Bez nich nebude automatické zobrazování fungovat, protože jsou realizovány ve skriptu „app.py“ který se stará o jejich interpretaci na displej. Takto navržený systém provázání dat v tabulkách, umožňuje členění zobrazovaných prvků a jejich znouvupoužitelnost v prvku nadřazeném.

4.2.6 Vytváření řídicího skriptu v jazyce Python

Poslední částí potřebnou pro funkci systému je program schopný dostupná data vhodně složit a odeslat do MCU převodníku. Pro tuto úlohu byly vytvořeny 2 sobě podobné skripty. První nese označení „app.py“, běží na pozadí systému a se stará o automatické zobrazování projektů na displeji, a druhý, který má za úkol jednorázově složit 1 bitový bitmapový obrázek a ten následně zobrazit na displeji. Oba jsou tak určeny k sestavení

bodů celého displeje, tedy vždy odesílají 140x19 zobrazovacích bodů (333 bajtů), a to pomocí sériové komunikace skrze „/dev/serial0“.

Vzhledem k tomu, že Python je skriptovací jazyk a jeho překlad neprobíhá při kompilaci programu v programovacím nástroji, ale při každém spuštění, je nutné do RPi přidat použité tzv balíčky (packages). V případě této práce se jedná o blíček „pymysql“, který se stará o komunikaci databází, a balíček „serial“ užívaný pro sériovou komunikaci s MCU. Samotnou instalaci popisuje např. zdroj [36].

- **Skript app.py**

Jelikož je potřeba aby zobrazování projektů probíhalo automaticky a po spuštění bylo dále nezávisle, běží tento skript na pozadí. Jeho běh je po spuštění nadále ovládán pomocí tzv. stavových souborů uložených ve složce „status“, která se pak nachází v kořenové složce internetových stránek, protože je ně spouštěn a zastavován. Druhou možností, kdy dojde k zastavení běhu programu je ukončení běhu OS. (například při výpadku napájení RPi)

Při spouštění je skript volán z internetových stránek skrze soubor projekty_start.php. Skript následně nastaví stav informující o běhu programu pomocí vytvoření souboru „script_running“. Následně si z databáze dbnastavení zjistí zvolený projekt a ten začne na displeji ve smyčce zobrazovat podle hierarchie popsané v podkapitole 4.2.5. Tento proces probíhá pomocí dotazování se databáze dbprojekty na strukturu a její data. Zeptá se tedy na pořadí displejů a jejich doby trvání. Pro každý displej se zeptá na jeho barvu a pořadí jednotlivých součástí. U nich zjistí pozici, barvu a obsah, a ten začne podle zjištěných parametrů skládat do výsledného displeje. Jakmile složí všechny součásti popsané pořadím označeným „idPoradiSoucasti“ v tabulce „soucasti“, vzniklý obraz se převede podle podkapitoly 4.1.4 a odešle ke zobrazení do MCU převodníku. Poté se vyčkává po dobu přiřazenou k danému displeji v kolonce „doba“ (hodnota je udávána v sekundách). Během ní se skript v pravidelných intervalech přesvědčuje, zda nebyl z řídicí internetové stránky „projekty.php“ (prostřednictvím souboru „projekty_stop“) vydán příkaz (vytvořen soubor) s názvem „stop_script“. Ten způsobí nastavení nového statusu pomocí odstranění souboru „script_running“ a následného vytvoření souboru „script_stopped“. Hned poté se programu ukončí. Celý tento proces (kromě inicializace) zahrnuje funkce s názvem „projekty_start“. V ní je pro převod dat do tvaru uvedeného v podkapitole 4.1.4 volána funkce „create_display“ a pro odeslání dat do MCU volána funkce „send_data“.

Ve výše popsáném běhu programu se vnořený proces skládání součástí do obrazovek ještě dělí na skládání textu a skládání obrázku. V případě skládání textu jsou jednotlivé obrazové body získávány pro každý znak z tabulky „txtfont_7x5“. Výsledkem je tak převod textu na pole zobrazovacích bodů. V případě potřeby, tak lze záměnou hodnot této tabulky změnit vzhled zobrazovaného textu. Činnost popsanou v tomto odstavci provádí funkce nazvaná „txt2polePix“.

Pro obrázky uložené v 1BPP formátu bitmapy je pak namísto skládání znaků z tabulky potřeba získat parametry a data v obrázku uložená. Struktura dat bitmapového obrázku je vysvětlena např. ve zdroji [37]. Pomocí něj lze z obrázku snadno získat potřebné parametry mezi něž patří jeho velikost a hodnoty barvy pro jednotlivé pixely. Nakonec je potřeba tyto hodnoty přeskádat do vhodného formátu zápisu pro funkci „create_display“. Tato funkce je v souboru uložena pod názvem „bmp2polePix“.

- **Skript „bmp.py“**

Jelikož oba soubory umožňují dekodovat a zobrazit na displeji 1BPP bitmapový obrázek jsou v této práci obsažené části funkcí „bmp2polePix“, „create_display“ a „send_data“. Ty jsou pak doplněny o načtení dat ze souboru s názvem „img_prime_ovladani.bmp“, který je uložen ve složce „img“ umístěné v kořenové složce řídicího webu.

Všechny vytvořené zdrojové kódy jsou dostupné v příloze G. Nachází se tam soubory webových stránek, oba napsané skripty a je zde vyexportována celá databáze užívaná při ověřování a testování funkčnosti.

5. DEMONSTRACE, OVĚŘENÍ A ZHODNOCENÍ

5.1 Příprava parametrů a testovacích dat

Před samotnou demonstrací bylo nejdříve potřeba nastavit pár přístupových údajů (jména a hesla), které je možno vidět v tab. 5.1.

Jméno	Heslo	Užívá
<i>Údaje pro přístup do databázi</i>		
webuser	webpassword	web pro dotazování dbprojekty, dbloginsys a aktualizaci dbnastaveni
scriptuser	scriptpassword	skript pro dotazování dbprojekty a dbnastaveni
admin21	password21	správce pro kompletní řízení a nastavení zobrazovaných dat
<i>Údaje pro přístup do internetových stránek</i>		
test	0000	uživatel pro řízení celého systému

Tab. 5.1 Tabulka přístupových údajů pro demonstraci funkce



Obr. 5.1 Ukázka obrazovek projekty „demo_bmp_PM“

Takto nastavené přístupové údaje slouží primárně pro účely demonstrace a testování výsledné funkce a neplní tak (pro svou jednoduchost, a hlavně vzhledem k jejich prozrazení) funkci ochrany před zneužitím neoprávněnou osobou.

Pro testování přímého řízení displeje pomocí textu byly do dvou vstupních kolonek vkládány textová data, které netřeba nijak zvláště připravovat. Pro testování přímého řízení pomocí vkládání obrázku bylo v programu GIMP2 vytvořeno několik testovacích bitmapových obrázků s 1BBP, a to v různých velikostech. Tyto obrázky jsou uloženy i v databázi dbprojekty. V té je navíc nahráno několik předpřipravených ukázek projektů, z nichž stojí za zmínku především ty označeny slovem „demo“. První je nazvána pouze „demo“ a zobrazuje kombinaci textů jako např. „FEKT“, „Vysoke uceni technicke“, aj. Díky tomu testuje skládání a zobrazování displejů složených z implementovaného fontu a textových dat. Druhá ukázka kombinuje vkládání textu s obrázkem a testuje tak navíc možnosti skládání a zobrazování textu v kombinaci s obrázkem. Poslední ukázka má název „demo_bmp_PM“ a zobrazuje „rychle“ (s odstupem 1 s) po sobě jdoucí obrázky pokrývající celou velikost displeje, čímž se testují možnosti zobrazování krátkých animací. Pro ukázku je

sada tvořící zmíněnou animaci zobrazena na obr 5.1. Celá databáze včetně všech vytvořených obrázků je obsažena v příloze G.

5.2 Demontrace a testování funkcí

Výsledný systém byl pomocí předpřipravených dat testován. Toto testování však probíhalo pouze na jediném z pěti zobrazovacích modulů, jak již bylo zmíněno, vlivem přijatých vládních omezení trvajících po většinu doby vymezené na vypracování této diplomové práce.

Výstupy z části demonstrace jsou z důvodů omezení maximální velikosti hlavního souboru a příloh z většiny popsány v následujícím textu. Ukázky jednotlivých obrazovek webových stránek je možné vidět v obrázcích v podkapitole 4.2.4.

5.2.1 Internetové stránky

Během testování přihlašování do stránek řízení displeje byly vyzkoušeny jejich reakce na vkládání různých přihlašovacích údajů:

- 1) V případě, že nejsou vyplněna obě přihlašovací pole je uživateli oznámeno: *“Vyplňte Uživatelské jméno a Heslo!”*,
- 2) V případě, že jsou přihlašovací údaje neplatné (neznámí uživatel, heslo nebo obojí) je uživateli oznámeno: *„Nesprávné přihlašovací údaje!“*
- 3) V případě, že uživatel zadá správné jméno i heslo, přihlásí se do systému a přejde na úvodní stranu.

Dále bylo ověřena správná funkce odhlášení, funkce navigační lišty a zobrazování jména přihlášeného účtu. K tomu bylo testováno náhodné vkládání adres funkčních souborů uložených ve složce „includes“. U žádné ze zmíněných funkcí nebyla objevena chyba.

Pro přímé ovládání zobrazování 2 řádků na displej bylo vyzkoušeno několik různých textů. Maximální délka textu vloženého do jednoho pole/řádku je omezena hned ve vstupním políčku na 26 znaků. Jelikož je ale pro překlad znaků do zobrazovaných bodů využito pouze znaků vyskytujících se v ASCII tabulce, jsou při zadávání jiných znaků zobrazovány náhodné obrazce. To je dané neošetřením tohoto chybového stavu.

Pro přímé ovládání pomocí vkládání obrázků mohou nastat tyto stavy:

- 1) V případě, že nebyl vybrán žádný obrázek zobrazí se hláška: *„Nejprve vyberte obrázek k nahrání!“*
- 2) V případě jiné přípony než „.bmp“ je zobrazena hláška: *„Nezle nahrát obrázek tohoto typu!“*.
- 3) V případě že má obrázek jiné rozměry než požadované (19x140pix) je zobrazena hláška: *„Rozměry obrázku neodpovídají velikosti displeje!“*
- 4) V případě že se vybere obrázek splňující rozměry a příponu, avšak s větší bitovou hloubkou, bude v případě velikosti nanejvýše 2500 bajtů nahrán ale při jeho zobrazení nebude dávat smysl. Jedná se o neošetřený stav.

- 5) V případě, že bude větší jak 2500 bajtů zobrazí se hláška: „Soubor je příliš velký!“

V části pojmenované projekty jsou správně zobrazovány názvy vybraných projektů a správně zobrazována všechna data v tabulkách včetně náhledů uložených obrázků. Pro zobrazení stavu jsou možné 3 stavové hlášky:

- 1) „*POZASTAVENO!*“ – pokud žádný projekt není zobrazován.
- 2) „*BĚŽÍ!*“ – v případě, že je projekt zobrazován
- 3) „*Chyba: stav nelze určit! Neexistuje žádný stav!*“ – se zobrazí v případě, že vyskytla neočekávaná chyba v běhu programu, běh programu byl při posledním běhu ukončen OS (například při odpojení napájení) apod. Chyba je způsobena nesprávnou signalizací stavu skriptu „app.py“

Výše zmíněná chyba může být opravena pomocí tlačítka, které se zobrazuje společně s chybovým hlášením. To spouští soubor projekty_reset.php a resetuje stavy a běh skriptu.

Pro úpravu dat uložených v databázi bylo využito funkcí phpMyAdmin. Tento způsob je sice funkční ale poněkud krkolomný a uživatel může snadno udělat chybu. Samotný databázový editor je zcela funkční, avšak nešlo se ubránit občasným chybovým hláškám spojeným s neznámou chybou.

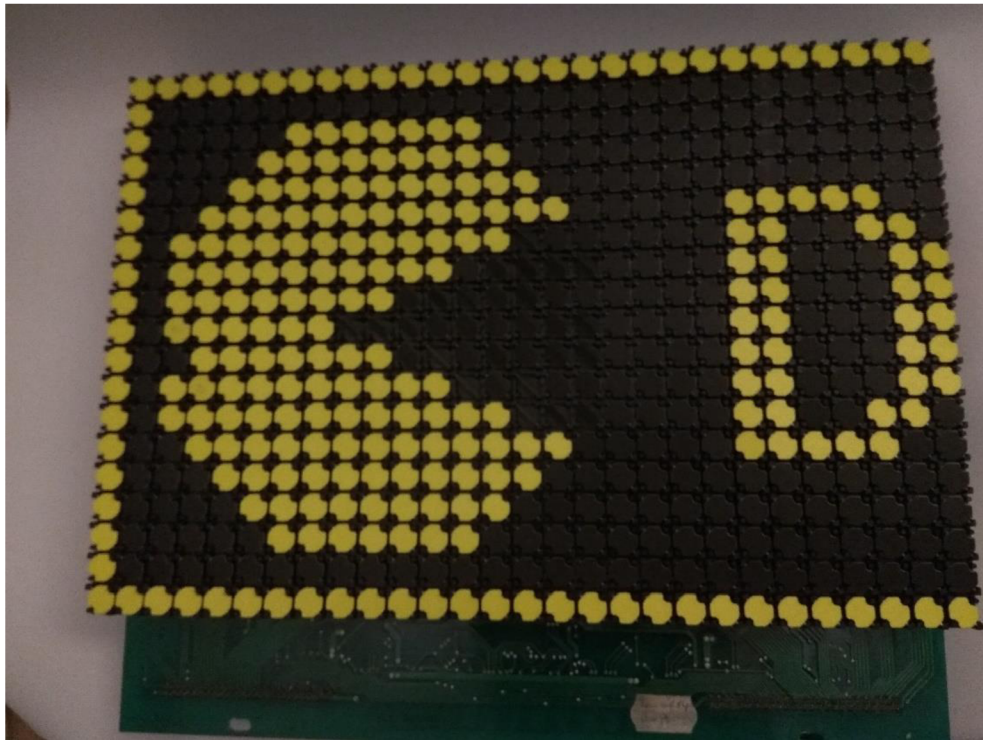
5.3 Výsledky zobrazení na displeji

Z důvodů popsaných v podkapitole 5.2. ani zde nelze vkládat všechny obrazové výstupy. Pro demonstraci je ale vhodné ukázat alespoň pár příkladů. Ty nejlépe ukážou funkci celého systému. Textové a obrazové výstupy na displej je možné vidět na obr. 5.3 a obr. 5.4.

5.4 Zhodnocení dosažených výsledků

Výsledky, které testování ukázalo, lze hodnotit kladně. Systém umí displej řídit v dostatečném rozsahu, tedy pomocí obrázků i textu. Pro řízení má uživatel dvě možnosti. První možností je zobrazit data přímo skrze tzv. přímé ovládání, a to jak textová data, tak i obrazová (obrázky). Při volbě druhé možnosti uživatel systém pouze iniciuje a ten pak již sám a automaticky zobrazuje displeje podle správně přednastavených parametrů.

Během testování se však ukázalo pár míst vhodných pro doplnění. Možnosti pro zlepšení jsou jasně vidět u návrhu projektů. To by si zasloužilo doplnit o vlastní uživatelské rozhraní v intuitivní grafické úpravě. Dále by bylo potřeba ošetřit některé neošetřené chybové stavy, kterými jsou třeba možnosti vkládání vstupů neodpovídajících udanému zadání. Například systém neumí rozpoznat bitovou hloubku vkládaného bitmapového obrázku. Tuto vadu lze omezit například i snížením maximální velikosti dat vkládaného obrázku. V neposlední řadě by bylo vhodné pro displej rozšířit znakovou sadu, aby bylo možné vkládat český text s diakritikou.



Obr. 5.3 Ukázka zobrazení obrazových dat



Obr. 5.2 Ukázka zobrazení textových dat

6. ZÁVĚR

V rámci této práce byla nejprve provedena literární rešerše na téma informační panely a zobrazovací jednotky.

Na ni navazuje analýza stávajícího flip-dot zobrazovacího panelu BUSE BS110, ve které byla identifikována většina obvodových prvků a jejich zapojení což přispělo k pochopení principu řízení displeje. Dále byla otestována funkčnost stávajícího HW a provedena oprava vadných bodů displeje v textu označených S1R1, S2R2 a S3R9 výměnou poškozených diod BAV99. U bodu S3R9 se tím ale nepodařilo chybu zcela odstranit. Měřením bylo zjištěno, že cívka tohoto pixelu vykazuje prokazatelně nižší hodnoty odporu než všechny ostatní. Na základě tohoto měření bylo usouzeno, že se v cívce vytvořil mezi závitový zkrat. K výměně poškozeného stripsu“ za nový však nebylo přistoupeno, a to z důvodů nedostatku náhradních dílů dostupných v dnešní době. Možnost opravy výměnou poškozeného vynutí cívky byla autorem zamítnuta z důvodu náročnosti provádění tohoto úkonu v domácích podmínkách. Opravu dalších 4 modulů displeje autor nemohl provést, jelikož ty zůstaly pro své nemalé rozměry v budově školy, a kvůli vládních opatření v reakci na šíření onemocnění COVID-19 je nebylo během doby trvání diplomové práce možno vyzvednout. Pro splnění dalších částí zadání práce již jediný dostupný modul plně postačoval, protože je jejich obvodové řešení identické a jediné v čem se při zapojení do jednoho celku liší je jejich adresa daná pořadím jejich propojení.

Po opravě displeje byl navržen a sestaven mikrokontrolerový systém skládající se navíc z řídicího mini počítače Raspberry Pi, který umožňuje jednoduché využití Ethernetového i bezdrátového rozhraní. Ten je dále připojen k mikrokontroléru, který řídí navržené obvody řízení napětí řádků a stávající obvody panelu BS110 pro řízení napětí jednotlivých sloupců.

Pro takto vzniklý systém bylo navrženo a vytvořeno SW vybavení umožňující zobrazování informací na připojeném panelu. Toho je dosaženo pomocí programového vybavení mikrokontroléru pro řízení obvodů displeje a komunikaci s nadřazeným mini počítačem Raspberry Pi. Pro ten se vytvořená SW výbava skládá z PHP webových stránek, MySQL databáze a obslužných programů psaných v jazyce Python.

Jak je možné vidět v kapitole 5, takto vzniklý systém byl v závěru práce také otestován na sadu testovací dat a parametrů, z nichž většina je součástí příloh práce. Z hodnocení tohoto testu vyplývá, že výsledné funkce zařízení splňují požadavky zadání. Jsou zde také zmíněny hlavní nedostatky současného stavu, mezi které patří složité zadávání dat tzv. projektů a při větším objemu uložených dat také jejich nízká přehlednost. Tento nedostatek by bylo vhodné v budoucnosti řešit rozšířením stránek o grafické uživatelské rozhraní, které by zjednodušilo vytváření obrazovek pomocí vnitřních vazeb a ty dále přímo ukládalo do vytvořené databáze. Další prostor pro

vylepšení objevené nedostatky v zabezpečení proti vkládání nepovolených dat a parametrů a v neposlední řadě umožnit vkládání znaků s diakritikou.

Literatura

- [1] *Portál herman.cz* [online]. [cit. 2020-05-05]. Dostupné z: <https://www.herman.cz/produkty/clanky-2/clanky/komunikacni-systemy/> HERMAN, Ivo. *Komunikační systémy ve veřejné dopravě*.
- [2] *Portál vcalc.net* [online]. [cit. 2019-11-16]. Dostupné z: <http://www.vcalc.net/display2.htm>. *Electronic Displays*.
- [3] *Portál circuitstoday.com* [online]. [cit. 2019-11-19]. Dostupné z: <http://www.circuitstoday.com/liquid-crystal-displays-lcd-working>. *Liquid Crystal Displays (LCD) – Working*.
- [4] *Portál* [online]. [cit. 2019-11-16]. Dostupné z: <http://en.fakoo.de/siekoo.html>. *The Siekoo Alphabet*.
- [5] *Portál plc-automatizace.cz* [online]. [cit. 2019-11-16]. Dostupné z: <http://plc-automatizace.cz/knihovna/hmi/rizeni.htm>. *Řízení displejů*.
- [6] *Portál plc-automatizace.cz* [online]. [cit. 2019-11-19]. Dostupné z: <http://plc-automatizace.cz/knihovna/hmi/technologie/mechanicke.htm>. *Elektromechanické displeje*.
- [7] *Portál pragotron.sk* [online]. [cit. 2019-11-19]. Dostupné z: https://pragotron.sk/?ukaz=content/is_tabule&=sk. *SIGNALTRON – TYPY INFORMAČNÝCH TABÚL*.
- [8] *Portál industrialalchemy.org* [online]. [cit. 2019-11-19]. Dostupné z: <https://www.industrialalchemy.org/articleview.php?item=1025>. *Staver 402L Vane Display*.
- [9] *Portál elektrorevue.cz* [online]. [cit. 2019-11-23]. Dostupné z: <http://www.elektrorevue.cz/clanky/99002/index.html>. *Moderní zobrazovací jednotky*.
- [10] *Portál buse.cz* [online]. [cit. 2019-11-16]. Dostupné z: <https://www.buse.cz/cs/zobrazovaci-prvky-flip-dot-a-flip-dot-led>. *Zobrazovací prvky "flip DOT" a "flip DOT-LED"*.
- [11] *Portál breakfastny.com* [online]. [cit. 2019-11-19]. Dostupné z: <https://breakfastny.com>. *BREAKFAST*.
- [12] *Portál i2.wp.com* [online]. [cit. 2020-01-02]. Dostupné z: <https://i2.wp.com/nuvitron.com/wp-content/uploads/2015/04/nuvitron-nixie-clock-replacement-tubes-on-geek-toys-isometric-view-in12-in-12.png>. *Nixie tubes*.
- [13] *Portál danyk.cz* [online]. [cit. 2020-01-02]. Dostupné z: <http://danyk.cz/digitrony.html>. *Digitrony*.
- [14] *Portál daliborfarny.com* [online]. [cit. 2019-12-02]. Dostupné z: <https://www.daliborfarny.com/>. *Moderní digitrony*.
- [15] *Portál herman.cz*. [online]. [cit. 2019-12-02]. Dostupné z: <http://www.herman.cz/cs/produkty/vybava/displej/vlp/vlp19/>. HERMAN, Ivo. *Vnější tabla – VLP 19/20*.

- [16] *Portál mobatime.cz* [online]. [cit. 2019-12-27]. Dostupné z: <https://mobatime.cz/obchod/lcd-informacni-tabule-rady-gl/>. *LCD informační panely řady GL*.
- [17] *Portál build-electronic-circuits.com* [online]. [cit. 2019-12-27]. Dostupné z: <https://www.build-electronic-circuits.com/h-bridge/>. *What Is an H-Bridge?*
- [18] *Portál digikey.com* [online]. [cit. 2019-11-19]. Dostupné z: <https://www.digikey.com/en/maker/blogs/2018/meet-the-new-raspberry-pi-3-model-b-plus>. *Meet the New Raspberry Pi 3 Model B+*.
- [19] *Portál raspberrypi.org* [online]. [cit. 2019-11-16]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/gpio/>. *GPIO*.
- [20] *Portál jsykora.info* [online]. [cit. 2019-11-19]. Dostupné z: <https://www.jsykora.info/2014/05/logic-voltage-levels/>. *Logic Voltage Levels*.
- [21] *Portál sparkfun.com* [online]. [cit. 2019-11-19]. Dostupné z: <https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide/all>. *Bi-Directional Logic Level Converter Hookup Guide*.
- [22] *Portál ti.com* [online]. [cit. 2020-01-02]. Dostupné z: <http://www.ti.com/lit/ds/symlink/l293d.pdf>. *L293x Quadruple Half-H Drivers*.
- [23] *Portál microchip.com* [online]. [cit. 2019-12-19]. Dostupné z: <http://ww1.microchip.com/downloads/en/devicedoc/doc2503.pdf>. *ATmega32*.
- [24] *Portál ti.com* [online]. [cit. 2019-12-19]. Dostupné z: <http://www.ti.com/lit/ds/symlink/cd54hc85.pdf>. *CD74HC85*.
- [25] *Portál ti.com* [online]. [cit. 2019-12-19]. Dostupné z: <http://www.ti.com/lit/ds/symlink/cd74hc138.pdf>. *CD54/74HC238*.
- [26] *Portál bourns.com* [online]. [cit. 2019-12-19]. Dostupné z: <https://www.bourns.com/pdfs/4800P.pdf>. *4800P Series - Thick Film Surface Mounted Medium Body*.
- [27] *Portál vishay.com* [online]. [cit. 2019-12-19]. Dostupné z: <https://www.vishay.com/docs/85718/bav99.pdf>. *Small Signal Switching Diode, Dual in Series*.
- [28] *Portál xx* [online]. [cit. 2019-12-19]. Dostupné z: <https://toshiba.semicon-storage.com/info/docget.jsp?did=4257&prodName=TD62083AFNG>. *Řízení*.
- [29] *Portál Toshiba.com* [online]. [cit. 2019-12-19]. Dostupné z: <https://toshiba.semicon-storage.com/info/docget.jsp?did=9829&prodName=TD62783APG>. *TD62083AFNG*.
- [30] *Portál avrfreaks.net* [online]. 2006 [cit. 2021-05-10]. Dostupné z: <https://www.avrfreaks.net/sites/default/files/gpio.h>. Dipl.-Ing. Heinrichs Hans-Juergen *GPIO.h*.
- [31] *Portál raspberrypi.org* [online]. [cit. 2021-05-10]. Dostupné z: <https://www.raspberrypi.org/documentation/installation/sd-cards.md>. *SD cards*.

- [32] *Portál howtoraspberrypi.com* [online]. [cit. 2021-05-11]. Dostupné z: <https://howtoraspberrypi.com/how-to-install-web-server-raspberry-pi-lamp/>. *How to install a web server on the Raspberry Pi*.
- [33] *Portál raspberrypi.org* [online]. [cit. 2021-05-11]. Dostupné z: <https://www.raspberrypi.org/documentation/linux/usage/commands.md>. *Linux commands*.
- [34] *Portál di-marco.net* [online]. [cit. 2021-05-11]. Dostupné z: https://di-marco.net/blog/it/2020-04-18-tips-disabling_bluetooth_on_raspberry_pi/. *Tips – Disabling Bluetooth on Raspberry Pi*
- [35] *Portál myhydropi.com* [online]. [cit. 2021-05-16]. Dostupné z: <https://myhydropi.com/install-phpmyadmin-on-a-raspberry-pi>. *Install PhpMyAdmin On A Raspberry Pi*
- [36] *Portál raspberrypi.org* [online]. [cit. 2021-05-16]. Dostupné z: <https://www.raspberrypi.org/documentation/linux/software/python.md>. *Installing Python packages*
- [37] *Portál medium.com* [online]. [cit. 2021-05-16]. Dostupné z: <https://medium.com/sysf/bits-to-bitmaps-a-simple-walkthrough-of-bmp-image-format-765dc6857393>. *Bits to Bitmaps: A simple walkthrough of BMP Image Format*

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

1BPP	1 bit pro pixel
apod	a podobně
atd	a tak dále
CCFL	Cold Cathode Fluorescent Light
CSS	Cascading Style Sheets
DPS	deska plošných spojů
FEKT	Fakulta elektrotechniky a komunikačních technologií
GPIO	General-purpose input/output
HW	hardware
IBIS	Integrated Bus Information System
IPIS	Integrovaný Palubní Informační Systém
LAMP	Linux, Apache, MySQL, PHP/Perl/Python
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
Obr	obrázek
PHP	Hypertext Preprocessor
RPi	Raspberry Pi
SDHC	Secure Digital High Capacity
SQL	Structured Query Language
SSH	Secure Shell
SW	software
tj	to je
tzv	takzvaný
VFD	Vacuum Fluorescent Display
VUT	Vysoké učení technické v Brně

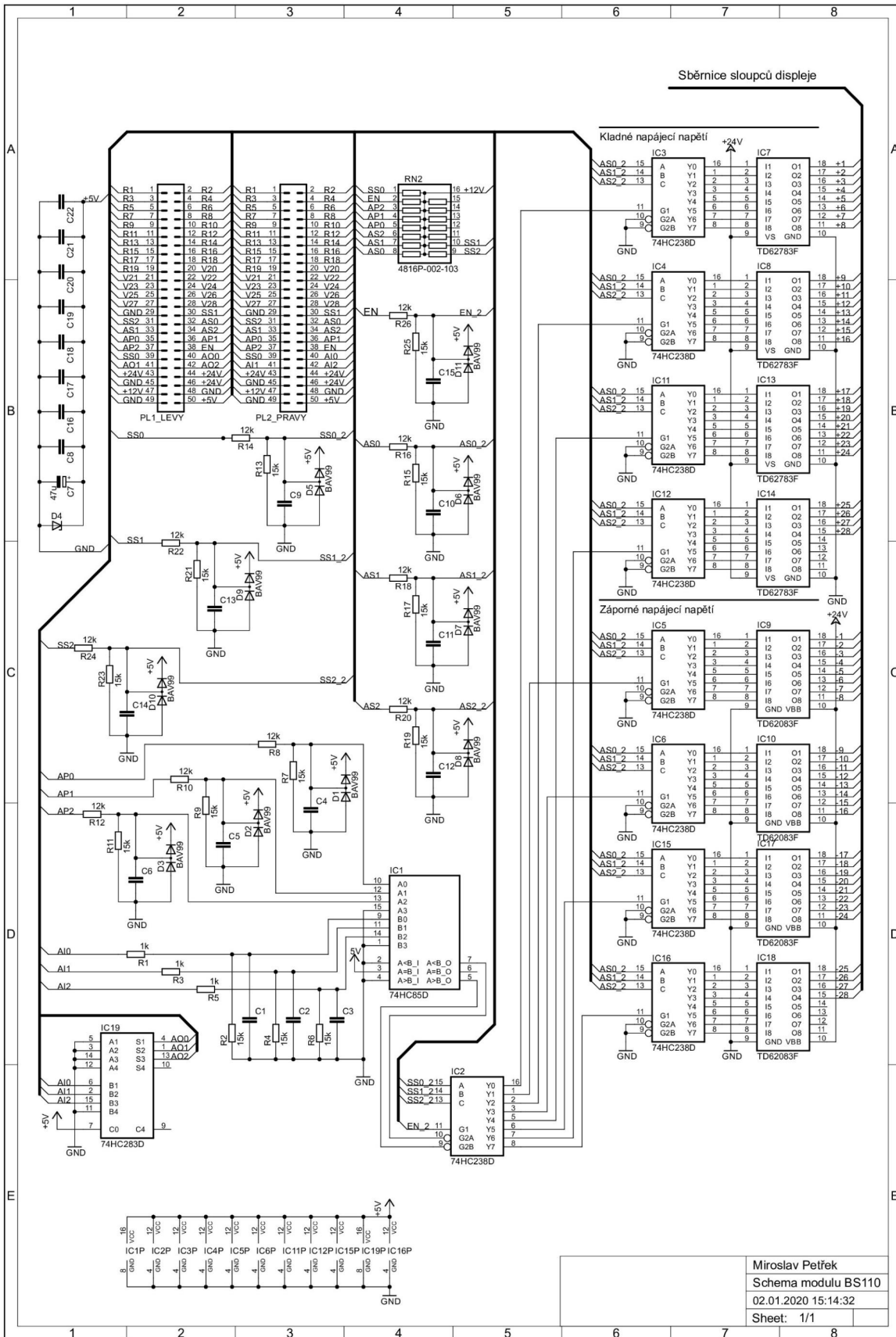
Symboly:

U	<i>napětí</i>	(V)
R	<i>odpor</i>	(Ω)
I	<i>proud</i>	(A)
t	<i>čas</i>	(s)
N_{del}	<i>hodnota děličky</i>	(-)
TC_{Vmax}	<i>maximální hodnota registru čítače</i>	(-)
TC_{Vset}	<i>přednastavená hodnota registru čítače</i>	(-)
F_{clk}	<i>frekvence připojeného krystalu (frekvence MCU)</i>	(Hz)
t_{ON}	<i>je čas potřebný pro vyvolání přerušení</i>	(s)

SEZNAM PŘÍLOH

PŘÍLOHA A - . ZREKONSTRUOVANÉ SCHÉMA ZAPOJENÍ MODULU BS110 PRO POCHOPENÍ FUNKCE	76
PŘÍLOHA B - NÁVRH ŘÍDÍCÍ ELEKTRONIKY VERZE 1.	77
PŘÍLOHA C - NÁVRH ŘÍDÍCÍ ELEKTRONIKY VERZE 2.	78
PŘÍLOHA D - PŘÍLOHA 4. PŘEDLOHA PRO VÝROBU DPS	79
PŘÍLOHA E - OSAZOVACÍ PLÁNY DPS	80
PŘÍLOHA F – SOUBORY NÁVRHU HW JSOU ULOŽENY NA PŘILOŽENÉM CD	
PŘÍLOHA G – SOUBORY NÁVRHU SW JSOU ULOŽENY NA PŘILOŽENÉM CD	
PŘÍLOHA H – SOUBORY PRO TESTOVÁNÍ SYSTÉMU JSOU ULOŽENY NA PŘILOŽENÉM CD	

Příloha A - Zrekonstruované schéma zapojení modulu BS110 pro pochopení funkce



Miroslav Petřek
Schema modulu BS110
02.01.2020 15:14:32
Sheet: 1/1

+5V ... Přiváděné napájecí napětí pro logické obvody

+12V ... Přiváděné napájecí napětí pro posílení signálových cest

+24V ... Přiváděné napájecí napětí pro ovládání cívek displeje

GND ... Zemní vodič / nulový potenciál všech zdrojů

Rx ... Řádek s číslem x (x je 1,2,3,...)

AO ... Address Out = A1+1(b), adresa vypočítaná pro modul připojený k levému konektoru

AI ... Address In - adresa přechozího modulu (z pravého konektoru)

APx ... Adresa Panelu - x. bit

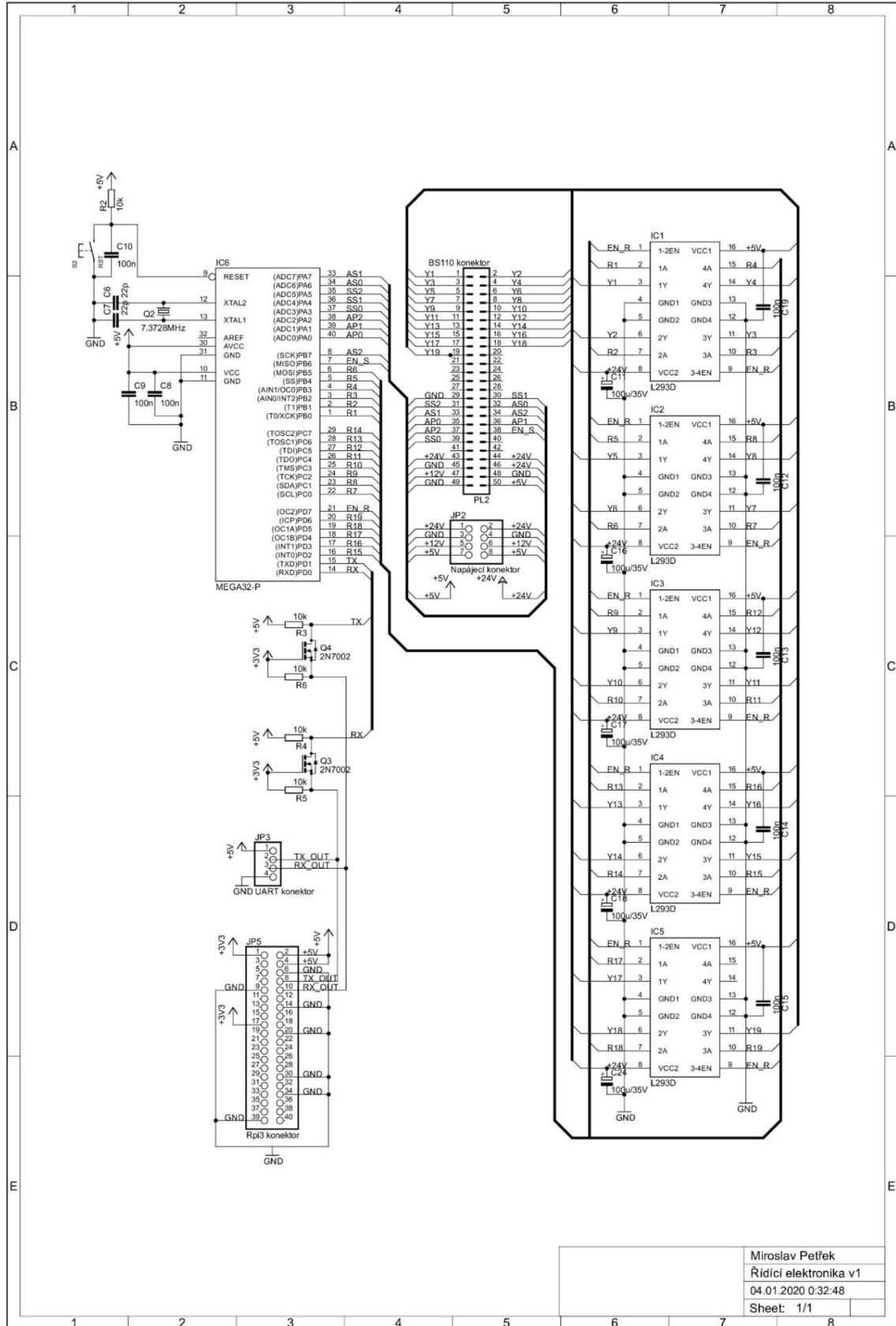
SSx ... adresa Skupiny Sloupců - x. bit

ASx ... Adresa Sloupce ve skupině - x. bit

XXx_2 ... Jsou řídicí signály po průchodu vstupním obvodem

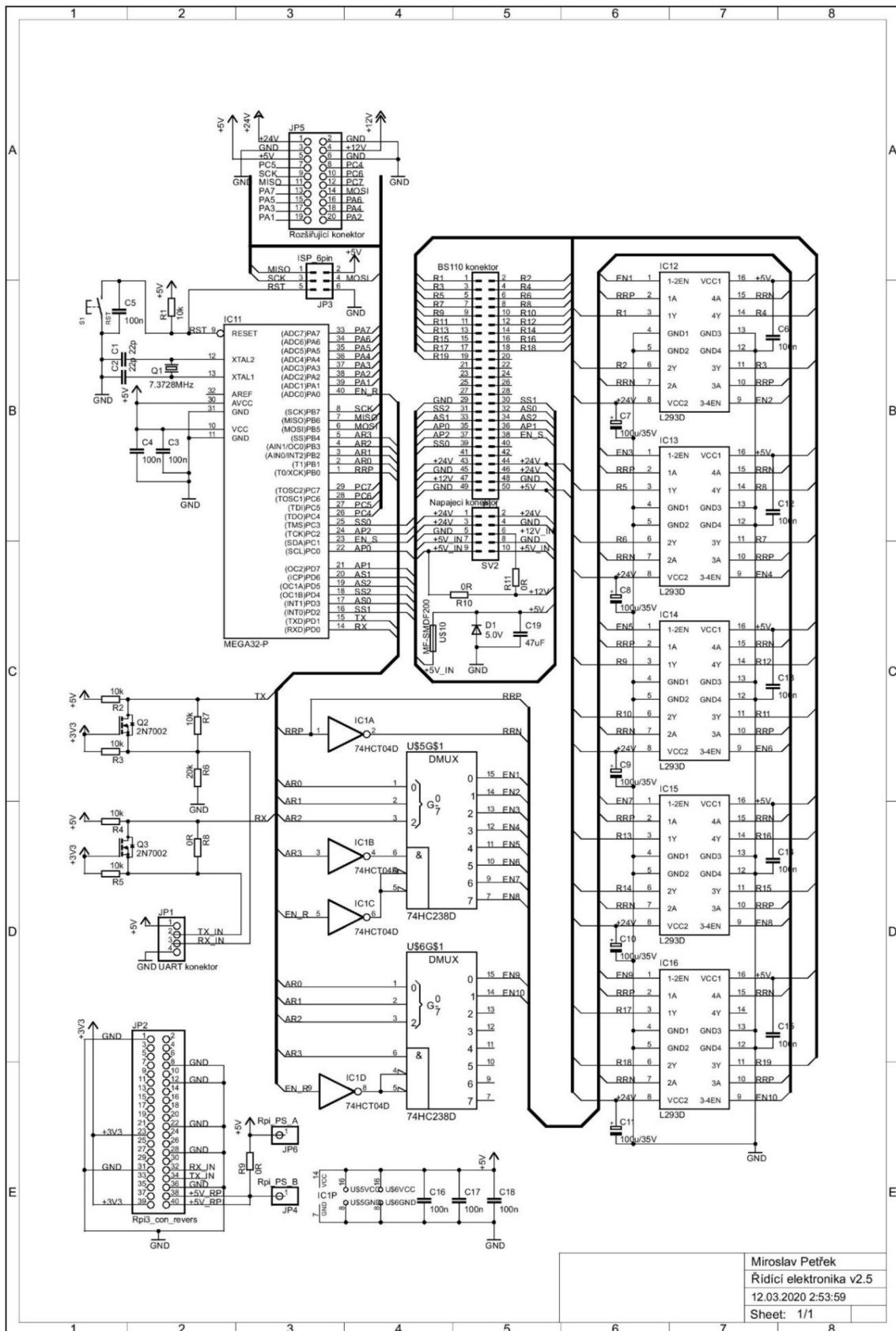
Vxx ... Vzájemně propojené vodiče tvořící oddělení mezi výkonovou částí řádků a logickým řízením sloupců

Příloha B - Návrh řídicí elektroniky verze 1.



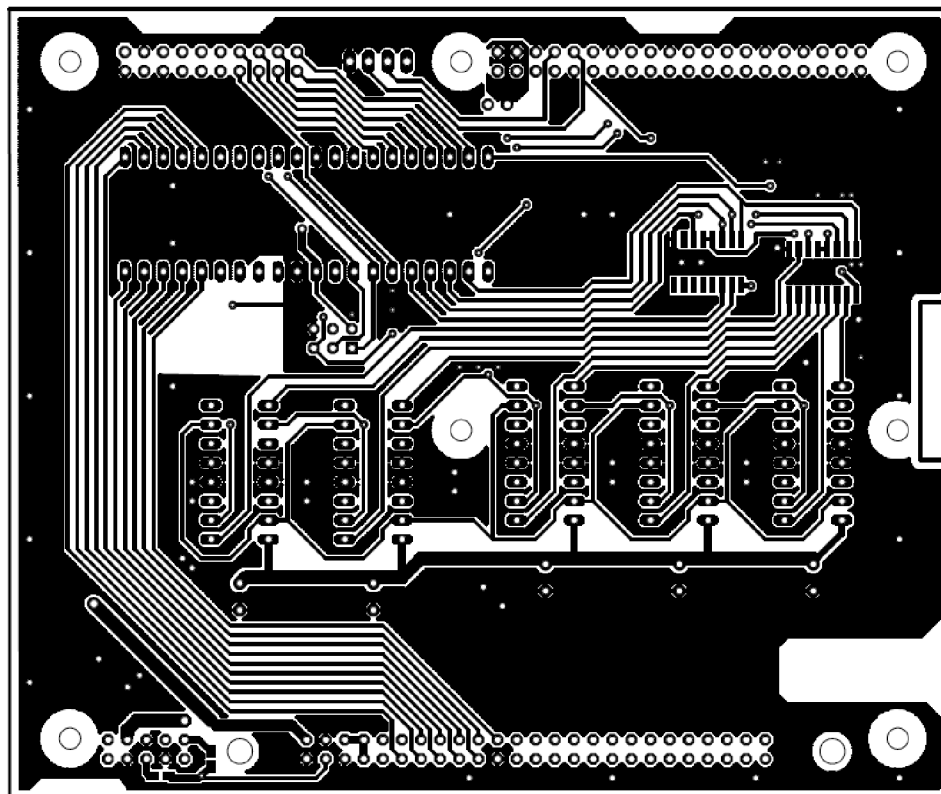
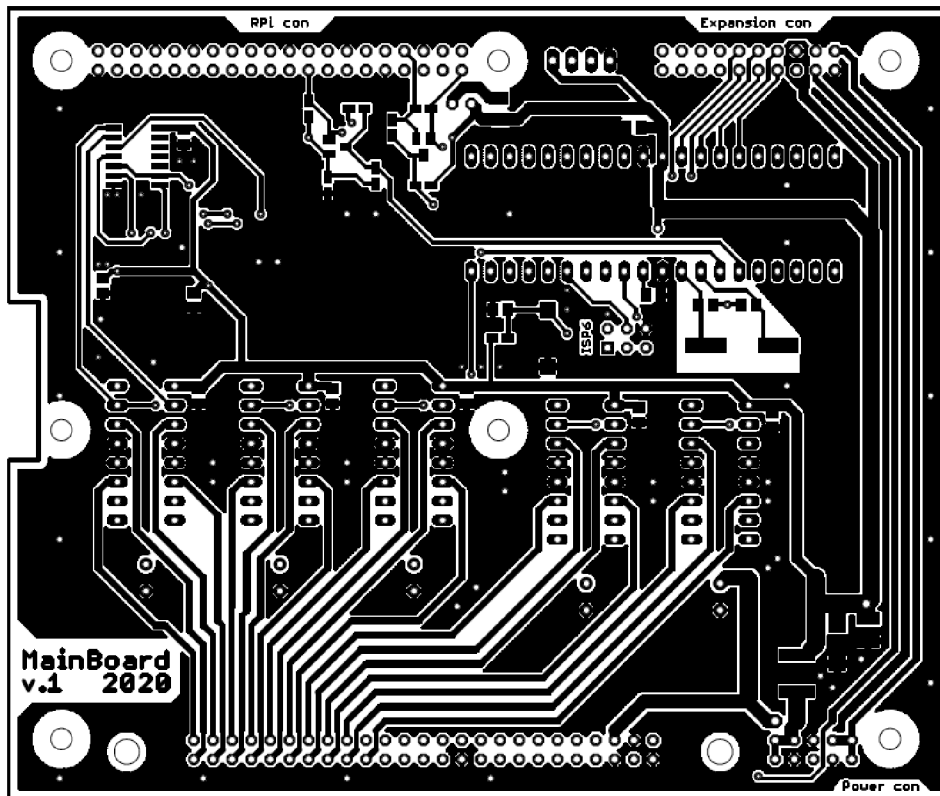
Miroslav Petřek
 Řídicí elektronika v1
 04.01.2020 0:32:48
 Sheet: 1/1

Příloha C - Návrh řídicí elektroniky verze 2.



Miroslav Petřek
 Řídicí elektronika v2.5
 12.03.2020 2:53:59
 Sheet: 1/1

Příloha D - Předloha pro výrobu DPS



Příloha E - Osazovací plány DPS

