



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

## ROZHRANÍ MEZI LCD MATICOVÝM A GRAFICKÝM DISPLEJEM

INTERFACE BETWEEN MATRIX AND GRAPHIC LCD

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Lukáš Hartmann

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Tomáš Götthans, Ph.D.

BRNO 2019



# Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**  
Ústav radioelektroniky

**Student:** Bc. Lukáš Hartmann

**ID:** 173652

**Ročník:** 2

**Akademický rok:** 2018/19

**NÁZEV TÉMATU:**

## Rozhraní mezi LCD maticovým a grafickým displejem

**POKYNY PRO VYPRACOVÁNÍ:**

Seznamte se s možnostmi maticových displejů. Prostudujte dostupnou literaturu. Zařízení bude nahrazovat používané LCD maticové displeje s řadičem HD44780 grafickým displejem. Bude nahrazovat používané displeje 2x20, 4x20 a 4x40. Navržené zařízení bude obsahovat tři druhy rozhraní. Prvním bude klasické paralelní rozhraní se stejným protokolem jako LCD maticové displeje. Dalším bude rozhraní UART, které bude sloužit ke konfiguraci displeje (nastavoval by se typ nahrazovaného displeje). Posledním použitým rozhraním bude SPI, sloužící pro další ovládání displeje. Navržené zařízení bude obsahovat plnou 16bitovou sadu UNICODE znaků. Bude zobrazovat vietnamštinu a arabštinu. Pro řízení bude použit procesor s architekturou ARM.

Navrhněte schéma zapojení, vyberte vhodný procesor, navrhněte a realizujte desku plošných spojů.

Oživate navrženou desku plošných spojů, vytvořte řídicí program pro procesor. Proveďte testování zařízení v praxi.

**DOPORUČENÁ LITERATURA:**

[1] Microchip Technology, MPLAB Harmony [online]. 2018 [cit. 2018-05-24]. Dostupné z: <https://www.microchip.com/mplab/mplab-harmony>

**Termín zadání:** 4.2.2019

**Termín odevzdání:** 16.5.2019

**Vedoucí práce:** doc. Ing. Tomáš Götthans, Ph.D.

**Konzultant:**

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

**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 částí druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Tato práce se zabývá návrhem rozhraní mezi alfanumerickým displejem s řadičem HD44780 a grafickým displejem. Toto zařízení nahrazuje displeje s rozlišením 2x20, 4x20 a 4x40 znaků. Znaky k zobrazení lze posílat paralelním rozhraním jako u alfanumerických displejů, nebo pomocí rozhraní SPI. Pro nastavení jazykové sady slouží komunikační linka RS-485. Jazykové sady jsou uloženy v externí paměti flash.

## **KLÍČOVÁ SLOVA**

Rozhraní mezi LCD maticovým displejem a grafickým displejem, ARM, LCD

## **ABSTRACT**

This work describes design interface between matrix with controller HD44780 and graphic LCD. This device replace 2x20, 4x20, 4x40 character display. Character which should be displayed are send through a parallel interface, such as matrix display, or send through SPI bus. Communication line RS-485 are used for set language character set. Language character sets are saved into external flash memory.

## **KEYWORDS**

Interface between LCD matrix display and graphic display, ARM, LCD

HARTMANN, L. *Rozhraní mezi LCD maticovým a grafickým displejem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2019. 30s. 12s příloh. Vedoucí diplomové práce Ing. Tomáš Götthans, Ph.D..

# PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Rozhraní mezi LCD maticovým displejem a grafickým displejem jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních anebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

# OBSAH

SEZNAM OBRÁZKŮ .....	VIII
ÚVOD .....	1
<b>1 POPIS ZAŘÍZENÍ .....</b>	<b>2</b>
<b>1.1 Co je systém APD1 .....</b>	<b>2</b>
1.1.1 Stanice dopravníku SD1.....	2
<b>1.2 Procesor .....</b>	<b>3</b>
1.2.1 Architektura ARM.....	3
1.2.2 Procesory ARM.....	4
1.2.3 ARM Cortex-M4.....	4
<b>1.3 Sériová komunikace .....</b>	<b>5</b>
1.3.1 USART/UART .....	5
1.3.2 RS-485 .....	5
1.3.3 Sériál peripheral interface – SPI .....	6
<b>1.4 LCD Displeje .....</b>	<b>7</b>
1.4.1 Alfnumerický displej.....	7
1.4.2 Grafický displej.....	8
<b>2 NÁVRH HARDWARE.....</b>	<b>9</b>
<b>2.1 Návrh zapojení.....</b>	<b>9</b>
<b>2.2 Návrh DPS.....</b>	<b>11</b>
<b>3 NÁVRH SOFTWARE.....</b>	<b>13</b>
<b>3.1 Knihovny pro obsluhu displeje.....</b>	<b>13</b>
3.1.1 St7920.h .....	13
3.1.2 RA6963.h .....	14
3.1.3 NT7108.h .....	14
<b>3.2 Knihovna pro práci s externí pamětí flash (flash_spi.h) .....</b>	<b>16</b>
<b>3.3 Knihovny pro změnu jazykové sady .....</b>	<b>16</b>
3.3.1 ST7920_spi.h .....	16
3.3.2 RA6963_spi.h .....	17
3.3.3 NT7108_spi.h .....	17
<b>3.4 Knihovny pro řadič HD44780 .....</b>	<b>17</b>

3.4.1	HD44780_P1.h .....	17
3.4.2	HD44780_P2.h .....	19
3.4.3	HD44780_P3.h .....	20
<b>3.5</b>	<b>Hlavní část programu .....</b>	<b>21</b>
3.5.1	RGD1_P1 .....	21
3.5.2	RGD1_P2 .....	23
3.5.3	RGD1_P3 .....	23
<b>3.6</b>	<b>Obslužný program pro zápis dat do flash paměti.....</b>	<b>23</b>
<b>4</b>	<b>TESTOVÁNÍ NAVRŽENÉHO ZAŘÍZENÍ .....</b>	<b>25</b>
<b>5</b>	<b>ZÁVĚR.....</b>	<b>27</b>
	LITERATURA .....	28
	SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK.....	30
	PŘÍLOHY .....	31

# SEZNAM OBRÁZKŮ

Obrázek 1.1: Stanice dopravníku SD1 .....	3
Obrázek 1.2: Srovnání procesorů s jádrem ARM .....	4
Obrázek 1.3: Zapojení linky RS-485 .....	6
Obrázek 2.1 : Napájení navrženého zařízení .....	10
Obrázek 2.2 : Zapojení vytvořené linky RS-485 .....	10
Obrázek 4.1 Grafický displej 192x32 bodů .....	25
Obrázek 4.2 Grafický displej 128x64 bodů .....	25
Obrázek 4.3 Grafický displej 240x64 bodů .....	26



# ÚVOD

Tato práce se zabývá návrhem zařízení pro náhradu alfanumerických displejů s řadičem HD44780 používaných u automatiky pásových dopravníků APD1 firmy Hansen Electric, spol. s r. o.. Jedná se o displeje s rozlišením 2x20, 4x20 a 4x40 zobrazovaných znaků. Tyto displeje jsou nahrazeny pomocí bodového grafického displeje. Použité displeje mají rozlišení 192x32, 128x64 a 240x64 bodů. Znaky k zobrazení se posílají pomocí paralelního rozhraní, jako u původních alfanumerických displejů. Pro komunikaci s ostatními zařízeními je využito rozhraní SPI. Pomocí něj lze také odesílat znaky k zobrazení. Dalším použitým rozhraním je UART (resp. linka RS-485), pomocí které se nastavuje jazyková sada k zobrazení na displeji. Jazykové sady jsou uloženy v externí flash paměti. Navržené zařízení obsahuje jazykovou sadu pro vietnamštinu a arabštinu.

Základem je mikroprocesor STM32L443 od firmy ST Microelectronic. Jedná se o 32 bitový systém s jádrem ARM Cortex-M4.

Tato práce, podobně jako má bakalářská práce, rozšiřuje možnosti systému APD.

# 1 POPIS ZAŘÍZENÍ

## 1.1 Co je systém APD1

Automatika pásových dopravníků (APD) je mikroprocesorový řídicí systém, který je primárně určen pro ovládání pásových dopravníkových linek v dolech nebo na povrchu. Jeho programové vybavení se stále rozšiřuje a nyní je možné pomocí tohoto systému již řídit celý chod porubu - od čerpacích stanic, kompresorovny, hřeblového dopravníku, drtiče materiálu, pluhu a dalších strojů. Provedení celého systému splňuje předpisy a nařízení pro provozování v důlních prostorách s nebezpečnými atmosférickými podmínkami a v prostorách s nebezpečím výbuchu metanu a uhelného prachu. Jedná se o jiskrově bezpečné zařízení.

Umožňuje přenos digitálních dat a analogového hovoru z dolu na povrch, vizualizaci procesu odtěžení, hlasité hovorové spojení, signalizaci provozních a havarijních stavů a vyslání varovných signálů před rozjezdem strojů v porubu nebo na důlních chodbách. Umožňuje také nouzové zastavení a blokování strojů. K dálkovému monitorování celého technologického procesu v dole je možné se k systému připojovat pomocí internetu a intranetu. Systém APD1 umožňuje také archivaci provozních stavů, naměřených hodnot i změny nastavení systémů a změn parametrů. Další vlastností tohoto systému je snadná diagnostika celého systému a lokalizace poruchy z ovládacího stanoviště na povrchu nebo v dole. Základní komponenty pro hlasové dorozumívání a vysílání akustických výstražných signálů jsou stanice dopravníku SD1 a hovorový zesilovač OPZ1. Celý systém je napájen jiskrově bezpečným napájecím zdrojem OZ12. Součástí systému jsou i různá čidla, snímače polohy, převodníky signálů a další prvky zajišťující kompletní technologický proces odtěžení v dole i na povrchu.

### 1.1.1 Stanice dopravníku SD1

Stanice dopravníku SD1 je vyrobená z ocelového plechu. Uvnitř stanice jsou umístěny elektrické obvody. Stanice dopravníku slouží k programovému řízení, ovládání a monitorování chodu dopravníkové linky včetně drtiče v dole i na povrchu. Umožňuje vizualizaci procesu odtěžení, hlasité hovorové spojení, signalizaci a vyslání, varovného signálu při rozjezdu strojů v porubu nebo na důlních chodbách. Umožňuje dále zastavení a blokování strojů na základě zpracování informací od stavů čidel a snímačů. Dále umožňuje programové ovládání výstupů pro řízení chodu strojů a komunikaci s další stanicí SD1 nebo s nadřazeným povrchovým počítačem. Stanici dopravníku je možno napájet jiskrově bezpečným zdrojem OZ12.

Hlasité hovorové spojení je realizováno pomocí mikrofону a reproduktoru umístěných na čelním panelu stanice dopravníku SD1. Uvnitř stanice dopravníku se nachází také akumulátor, který je za provozu dobíjen ze zdroje OZ12. Tento akumulátor vykrývá při hovoru nebo signalizaci zvýšenou spotřebu elektrické energie pro napájení zesilovače s reproduktorem. Na čelním panelu skříně jsou umístěna ovládací tlačítka, signalizační LED diody, blokovací tlačítko a průzor, pod nímž se nachází alfanumerický LCD displej informující o provozních stavech systému. Ke spodní části

pláště krabice je šrouby přichycena výměnná deska s konektory. V základním provedení stanice dopravníku je na připojovací desce 31 konektorů pro připojení čidel a snímačů, jedna vývodka a dvě sedmipólové zásuvky pro připojení speciálních propojovacích kabelů. [1]



Obrázek 1.1: Stanice dopravníku SD1

## 1.2 Procesor

### 1.2.1 Architektura ARM

Architektura ARM (Advanced RISC Machines), vyvinutá v Británii firmou ARM Limited, je založena na filozofii RISC. První mikroprocesor byl navržen v roce 1984 [2]. V dnešní době patří mezi nejrozšířenější procesory v počtu vyráběných kusů. Používají se v mnoha zařízeních spotřební elektroniky pro různé smartphony, tablety a herní konzole. Dále se používají ve vestavěných systémech jako např.: Wi-Fi čipy, routery apod. Vyznačují se nízkou spotřebou energie při vysokém výpočetním výkonu, proto jsou vhodné pro bateriově napájené zařízení. Díky menší spotřebě se dají využít i v náročných tepelných podmínkách. Také nepotřebují složité chlazení, které může být nespolehlivé [3].

Základní charakteristika:

- 32-bitová architektura i datová sběrnice
- tzv. Load/Store architektura – všechny operace jsou prováděny pouze mezi registry. Pokud se má pracovat s daty uloženými v paměti, musí se tato data nejprve přenést do registru. Poté se mohou data opět zapsat do paměti.

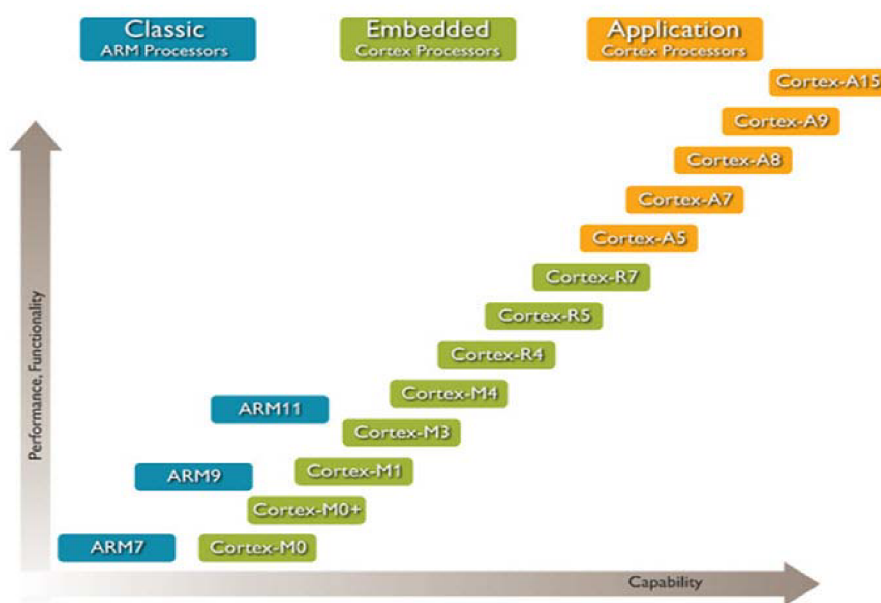
- Čtyři základní režimy:
  - Uživatelský,
  - Přerušení s vysokou prioritou,
  - Přerušení s normální prioritou,
  - Supervisorský – po resetu

## 1.2.2 Procesory ARM

V současné době existují dvě hlavní rodiny procesorů. Jedná se o rodinu Cortex-M, která je určena pro mikrokontroléry. Další rodinou je Cortex-R, která poskytuje vysoký výpočetní výkon, pracující v reálném čase.

Existuje také rodina Cortex-A, která se zaměřuje na vysoký výkon v aplikacích, jenž vyžadují platformní operační systém. [4]

Na následujícím obrázku je srovnání vyráběných procesorů s jádrem ARM (převzato z [4]).



Obrázek 1.2: Srovnání procesorů s jádrem ARM

## 1.2.3 ARM Cortex-M4

Jedná se o procesor s Harvard architekturou. Výhodou této architektury je rozdělená paměť pro data a pro instrukce. V jeden okamžik může číst procesor data i instrukce, což zvyšuje výpočetní výkon.

Procesory s jádrem Cortex-M4 jsou převážně určeny do aplikací ve kterých je potřeba digitálního zpracování signálu nebo pro práci s daty s plovoucí řádovou čárkou.

Základní instrukce zpracovávané jádrem Cortex-M4 jsou shodné s čipy *Cortex-M3* i *Cortex-M7*, protože je podporována jak instrukční sada Thumb s šestnáctibitovými operačními kódy, tak i instrukční sada Thumb-2, v níž kromě šestnáctibitových operačních kódů nalezneme i instrukce s 32bitovým slovem. Naproti tomu původní „čistě RISCová“ 32bitová instrukční sada ARM podporována není.[5]

Vlastností tohoto procesoru s jádrem Cortex-M4 je jednotka pro správu paměti (MPU). Paměť může být rozdělena do regionů a každé části lze přiřadit jiné právo přístupu.

## 1.3 Sériová komunikace

Často se používá pro přenos dat mezi různými zařízeními. Jedná se o přenos dat po jednotlivých bitech po sběrnici.

Dělí se na synchronní a asynchronní přenos. Rozdíl je v použití hodinového signálu a v počtu potřebných vodičů.

Synchronní přenos je tedy řízen synchronizačními hodinovými pulzy. Na začátku každého přenášeného slova jsou ještě navíc vysílány synchronizační pulzy. Vysílač nepřetržitě posílá data. Pokud nejsou žádná data k odeslání, vysílá pouze synchronizační impulzy. Tento přenos je např. využíván při programování mikrořadičů PIC [6].

Asynchronní přenos probíhá bez synchronizačních hodinových impulzů. Doba trvání je určena přenosovou rychlostí. Komunikující zařízení tedy musí mít stejně nastavenou přenosovou rychlost. Začátek vysílání začíná synchronizačním impulzem, který je tvořen přechodem z vysoké úrovně do nízké. Poté následují datové bity (5 až 8) a dále může vysílaný řetězec obsahovat paritní bit pro kontrolu dat. Nakonec následuje stop bit. Data mohou být vysílána v libovolných intervalech (doba mezi jednotlivými vysíláními může být různá) [7].

### 1.3.1 USART/UART

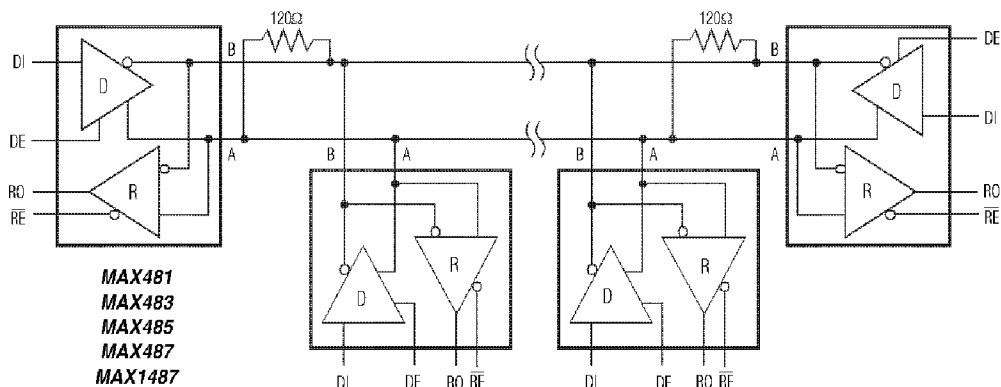
USART, resp. UART je synchronní, resp. asynchronní sériové rozhraní pro obousměrný datový přenos (plný duplex). Data jsou odesílána pomocí pinu označovaného TX (transmit) a přijímána na pinu RX (receive).

Pokud neprobíhá vysílání, je klidová úroveň definována jako logická 1. Vysílání je zahájeno tzv. start bitem, což je reprezentováno přechodem z log. 1 do log. 0. Dále následuje 5 až 9 datových bitů, které jsou odesílány v pořadí od nejméně významného (LSB) po nejvýznamnější (MSB). Datové bity jsou následovány volitelným paritním bitem a jedním až dvěma stop bity. Stop bity jsou reprezentovány přechodem logické úrovně z 0 do 1. Po stop bitu lze odeslat další sekvenci dat. Datová sekvence je odesílána zvolenou symbolovou rychlostí. Pokud během přenosu došlo k poškození dat, jsou přijímací jednotkou signalizovány různé chyby. Může to být například chyba parity nebo chyba rámce (pokud nesouhlasí počet stop bitů).

### 1.3.2 RS-485

Jedná se o komunikační standard, který se používá především v průmyslovém prostředí. Je podobný standardu RS-232, od kterého se liší především úrovněmi napětí. Linka RS-485 používá pro oba směry přenosu dat dva kroucené vodiče označované A a B, které přenášejí symetrický signál. Komunikace je poloduplexní, je nutné zajistit přepínání směru vysílání a přijímání. Na linku lze připojit až 32 vysílačů. Při komunikaci na větší vzdálenosti (maximálně 1200m) je nutné použít zakončovací odpory tzv. terminátory. Jsou určeny k potlačení odrazů na vedení. Doporučená hodnota je 120Ω

(viz obr. 1.2).[8] Také je nutné definovat klidovou úroveň na jednotlivých linkách. K tomu stačí propojit vodič linky přes rezistor na napájecí napětí. Definování klidové úrovně je důležité, aby nedošlo vlivem šumu k nežádoucímu příjmu dat. Aby byla zajištěna funkčnost sběrnice, musí se všechny přijímače i neaktivní vysíláče nacházet ve stavu vysoké impedance.



Obrázek 1.3: Zapojení linky RS-485

### 1.3.3 Seriál peripheral interface – SPI

Jak již název napovídá, jedná se o sériové komunikační rozhraní. Pomocí tohoto rozhraní lze propojit dva nebo více uzlů (zařízení). Jeden uzel je v režimu master (většinou mikrokontrolér), ostatní uzly jsou v režimu slave. Master řídí veškerou komunikaci pomocí hodinového signálu, který generuje na vodiči označovaném symbolem SCK.[9] Propojení uzlu master a slave hodinovým signálem zajistí zcela synchronní přenos dat. Pro obousměrný přenos dat je nutné propojit vodiče s označením MISO (Master In Slave Out) a MOSI (Master Out Slave In). Dalším používaným vodičem je vodič s označením SS (Slave Select). Ten slouží pro výběr konkrétního uzlu pro komunikaci. V případě komunikace pouze dvou uzlů je vodič SS nepotřebný. Naopak při komunikaci více uzlů je potřeba propojit vodič SS od každého uzlu s jedním pinem mikrokontroléru.

Díky své jednoduchosti (komunikační rozhraní SPI využívá pouze jednosměrné porty) se rozhraní využívá v mnoha systémech. Příkladem jsou některé typy pamětí, A/D a D/A převodníky, obvody reálného času (RTC), atd.

Hlavní výhodou, jak již bylo zmíněno, je jednoduchost implementace. Další výhodou může být jednoduchost protokolu. Ve své podstatě totiž není SPI nic jiného než několik externě propojených posuvných registrů, přičemž posun je řízen jednotným hodinovým signálem. Jednoduchost přenosového protokolu vyplývá i z toho, že data jsou přenášena obousměrně po samostatných vodičích, takže není nutné řešit přepínání mezi vysíláním a příjmem jako tomu bylo u některých sběrnic používajících poloduplexní přenos dat. Hodinová frekvence, z níž se odvozuje bitová rychlost (bitrate), se v běžných případech pohybuje v rozsahu 1 MHz až 10 MHz. Při dodržení dostatečně malé kapacity spoje je však možné použít i poměrně vysokou frekvenci až 70 MHz. [9]

Hlavní nevýhodou je existence pouze jednoho zařízení pracujícího v režimu master. Komunikace může probíhat také v tzv. režimu multiple master, ale je potřeba složitější

přenosový protokol. V tomto případě se vytrácí hlavní výhoda rozhraní SPI, a to je její jednoduchost.

## 1.4 LCD Displeje

Displej slouží k zobrazování informací srozumitelných pro uživatele. LCD displej je tenké ploché zobrazovací zařízení, které se skládá z barevných nebo monochromatických pixelů. Ty jsou seřazeny před zdrojem světla nebo reflektorem.

Každý pixel LCD displeje se skládá z molekul tekutých krystalů uložených mezi dvěma průhlednými elektrodami a mezi dvěma polarizačními filtry, přičemž osy polarizace jsou na sebe kolmé. Bez krystalů mezi filtry by bylo světlo procházející jedním filtrem blokováno filtrem druhým. Molekuly tekutých krystalů jsou bez elektrického proudu v chaotickém stavu. Elektrický proud způsobí, že se molekuly srovnají s mikroskopickými drážkami na elektrodách. Drážky na elektrodách jsou vzájemně kolmé, takže molekuly se srovnají do spirálové struktury (onen krystal). Světlo procházející filtrem je při průchodu tekutým krystalem rotováno, což mu umožňuje projít i druhým filtrem.[10]

Existuje celá řada LCD displejů, které se všeobecně dělí na dva typy. První z nich rozděluje LCD displeje do dvou skupin – alfanumerické (znakové) a grafické. Druhý typ dělení je podle barvy - barevné a jednobarevné (monochromatické).

### 1.4.1 Alfanumerický displej

Alfanumerické displeje dokáží zobrazovat pouze písmena a číslice. Lze zobrazit i jednoduché grafické prvky, které jsou definované v základním rastru znaků. Výhodou těchto displejů oproti grafickým displejům je jednoduchý zápis k zobrazení. Znaky jsou předdefinovány v paměti displeje.

Obrazovka displeje je rozdělena na  $M \times N$  políček, kde každé políčko je definováno pomocí  $m \times n$  bodů. Každý řádek textu na obrazovce je proto tvořen  $m$  rastrovými řádky. Displej nejprve vykresluje všechny body jednoho rastrového řádku určitého znaku, potom vykresluje tentýž rastrový řádek znaku o jednu pozici doprava, atd. Tak se pokračuje až po dosažení konce řádku. Potom pokračuje dalším rastrovým řádkem stejných znaků. Až dosáhne posledního rastrového řádku na daném textovém řádku, následuje přechod na další textový řádek.[11]

Většina alfanumerických displejů pracuje s řadičem HD44780. Vyrábí se v rozlišení od 1x8 až po 4x40 znaků. K mikrokontroléru jej lze připojit pomocí čtyř nebo osmi datových vodičů a třech řídicích vodičů: R/S – pro přepínání zápisů instrukcí nebo dat, E – signál pro zápis dat do řadiče a R/W – pro zápis nebo čtení. Displeje především slouží k zobrazení dat pro obsluhu, tzn. nečteme z nich data, proto může být signál R/W trvale uzemněn. Řadič HD44780 může zobrazovat pouze 80 znaků, což vystačí pouze na displeje s maximálním rozlišením 2x40 a 4x20 znaků. Displeje s rozlišením 4x40 znaků proto mají dva řadiče, kde každý ovládá pouze polovinu. Tomu je i uzpůsobeno ovládání, kde je navíc jeden signál E a ostatní signály jsou zapojeny paralelně k oběma řadičům. Řadiče HD44780 také umožňují doplnit znakovou sadu osmi vlastními znaky.

## 1.4.2 Grafický displej

Grafické displeje pomalu nahrazují používané alfanumerické displeje. Z důvodu nižší ceny se většinou používají monochromatické displeje. Displeje se vyrábí k použití ve velkém teplotním rozsahu. V komerčním rozsahu se teploty pohybují obvykle od 0 do +50°C a v průmyslovém od -20 až po +70°C. Při nižších teplotách displeje zpomalují svoji reakci na změnu zobrazované informace. Může se stát, že displej „zamrzne“ a trvale bude zobrazovat poslední informaci. Tento proces je obvykle vratný. Pokud se displeje používají nad horní hranici teplotního rozsahu, klesá kontrast, až displej úplně zčerná a je nečitelný. [12]. Některé grafické displeje obsahují budič záporného napětí, které je potřeba pro nastavení kontrastu.

Grafické displeje jsou řízeny řadiči s různým stupněm inteligence. Displej je v podstatě oblast malých bodů, které buď svítí, nebo nesvítí. Rozlišení se udává v pixelech, což je počet bodů v řádcích a sloupcích (např. 168x64). Velikost jednoho bodu udává jemnost zobrazení. Rozsvícení jednotlivých bodů ve většině případů zajišťuje řadič displeje, který je integrován na plošném spoji. Často užívanými řadiči jsou např. KS107 a KS108 od firmy Samsung, které umožňují obsluhovat displeje do velikosti 128 x 64 bodů, a Toshiba T6963 obsluhující displeje od 128 x 64 bodů až po 240 x 128 bodů. Pro displeje s větším rozlišením se používá např. EPSON SED1335, který obsluhuje displeje až do rozlišení 640 x 256 bodů. Z důvodu snížení ceny displejů pro odběratele se vyrábějí i displeje bez řadičů, kde řízení všech bodů displeje musí zajistit externí elektronická jednotka [12].

### 1.4.2.1 Řadič ST7920

Tento řadič podporuje tři druhy rozhraní. Jedná se o paralelní rozhraní, které může být 8 bitové nebo 4 bitové a sériové rozhraní. Řadič podporuje znakový i grafický mód. Pokud je zvolen znakový mód lze zobrazit znaky ze dvou různých znakových sad. První znaková sada obsahuje čínské znaky s rozlišením 16 x 16 bodů. Druhá sada obsahuje alfanumerické znaky s rozlišením 16 x 8 bodů. V grafickém módu lze ovládat zobrazitelnou plochu s maximálním rozlišením 64 x 256 bodů.

Řadič ST7920 lze použít v širokém rozsahu napájecího napětí (2,7V – 5,5V). Další výhodou je nízká spotřeba. Tento obvod je tedy vhodný pro bateriově napájené zařízení [13].

### 1.4.2.2 Řadič NT7108

Řadič NT7108 je ovládán pomocí 8 bitového paralelního rozhraní. Neobsahuje žádnou znakovou sadu, proto se jedná o čistě grafický displej. Jeden řadič dokáže ovládat až 64 x 64 grafických bodů, přestože je používán u displejů s větším rozlišením až 128 x 64 bodů. Proto se u větších displejů zapojují paralelně a každý řadič ovládá pouze polovinu displeje. Napájecí napětí se může pohybovat v rozmezí 2,7V – 5,5V [14].

### 1.4.2.3 Řadič RA6963

Tento řadič podporuje jak grafický tak i znakový mód. Podporuje také smíšený režim (text + grafika). Textový mód obsahuje 256 znakovou paměť ROM pro ASCII znaky, japonské znaky a číselné znaky. Řadič RA6963 má 8 bitovou paralelní datovou sběrnici, která může být přímo připojena k řídicímu mikroprocesoru. [15]. je plně kompatibilní s řadičem T6963C



## 2 NÁVRH HARDWARE

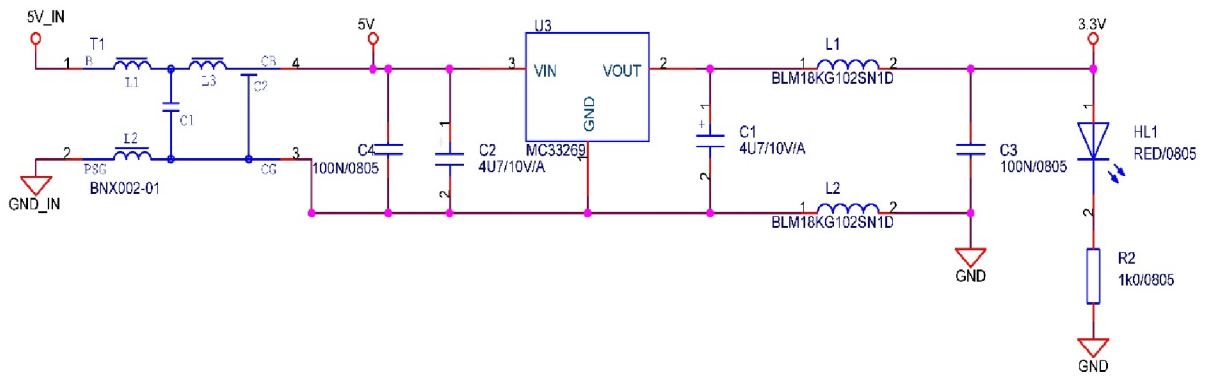
Nahrazované alfanumerické displeje jsou různě velké a proto i grafické displeje byly zvoleny s třemi různými rozlišeními. Alfanumerický displej s rozlišením 2x20 znaků bude nahrazen grafickým displejem s rozlišením 192x32 bodů. Tento displej ovládá řadič ST7920. Dalším zvoleným displejem je displej s rozlišením 128x64 bodů pro náhradu alfanumerického displeje s rozlišením 4x20 znaků. Řízení tohoto displeje obstarává řadič NT7108. Poslední nahrazený displej má rozlišení 240x64 bodů pro náhradu alfanumerického displeje s rozlišením 4x40 znaků. Pro každý nahrazovaný displej byla navržena samostatná DPS. Rozměry navržených DPS byly zvoleny s ohledem na rozměry displejů.

### 2.1 Návrh zapojení

Jak již bylo zmíněno byly navrženy tři různé DPS. Jejich hlavní rozdíl spočívá v propojení mikroprocesoru s grafickým displejem. Displej s rozlišením 192x32 bodů od firmy Newhaven display international používá pro komunikaci osm datových pinů a tři řídicí piny (RS, R/W, E). RS (register select) určuje, zda jsou posílána data nebo instrukce. R/W – (Read/write) určuje, zda jsou data z displeje čtena nebo jsou data zapisována. E (Enable) potvrzení zvoleného příkazu. Displej s rozlišením 128x64 bodů od firmy Raystar optronics používá pro komunikaci osm datových a šest řídicích pinů (D/I, R/W, E, CS1, CS2, Reset). Pin D/I (Data/Instruction) určuje, zda jsou odesílána data nebo instrukce. Pin CS1 a CS2 slouží k výběru sloupce (CS1 vybírá sloupce 1 – 64, CS2 vybírá sloupce 65 - 128). Pokud je pin v logické 1 jsou dané sloupce vybrány. Pin RESET provede reset celého displeje. Poslední zvolený displej má rozlišení 240x64 bodů od výrobce Raystar optronics. Displej používá pro komunikaci osm datových a šest řídicích pinů (WR, RD, CE, C/D, Reset, FS). Pokud je pin WR v logické 0 zapisují se do displeje data. Pin RD v logické 0 umožní číst data z displeje. Pin CE v logické 0 povoluje komunikaci s řadičem v displeji. Pin C/D (Command/Data) určuje, zda má řadič RA6963 data/instrukce číst nebo zapisovat na port.

Řízení navrženého zařízení obstarává 64 pinový mikroprocesor od firmy ST microelectronics STM32L443. Jedná se o low-power mikroprocesor s architekturou ARM Cortex-M4. Programování probíhá pomocí sériového programovacího rozhraní SWD.

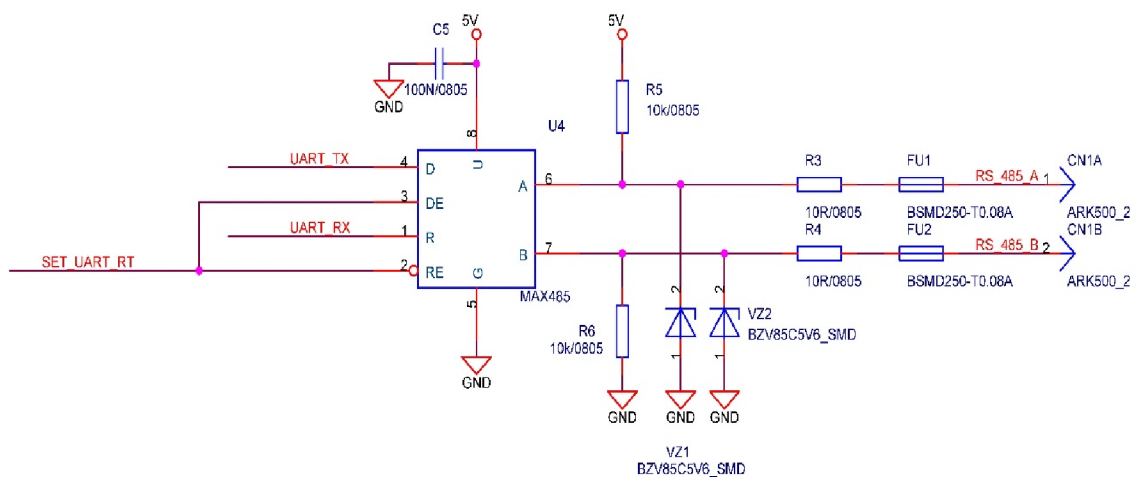
Napájení zařízení je řešeno externím zdrojem. Přivedené napájecí napětí je 5V, které je filtrováno pomocí odrušovacího filtru BNX002-01. Mikroprocesor požaduje napájecí napětí 3,3V. Toto napětí vytvoří stabilizátor MC33269. Pro lepší odrušení napájecího napětí jsou za stabilizátor zapojeny odrušovací tlumivky. Viz následující obrázek:



Obrázek 2.1 Napájení navrženého zařízení

Dle doporučení výrobce mikroprocesoru jsou u napájecích pinů zapojeny blokovací kondenzátory. Pro komunikaci mikroprocesoru s okolím se využívá komunikační rozhraní UART a SPI.

Rozhraní UART je připojeno k převodníku MAX485CSA od firmy Maxim Integrated. Převodník vytvoří komunikační linku RS-485. Úkolem tohoto obvodu je změnit napěťové úrovně z mikroprocesoru (0 a 3,3V) na 0 a 5V. Na vytvořenou linku RS-485 jsou zapojeny rezistory s hodnotou 10kΩ pro definování klidové úrovně linky. K ochraně před napěťovými špičkami slouží zenerovy diody BZV55C5.6V. Další ochrana linky je vytvořena pomocí pojistek BSMD250 s vybavovacím proudem 80mA a 10Ω rezistorů. To zajistí ochranu při přetížení a případném poškození integrovaného obvodu při zkratu. Zapojení převodníku na linku RS-485:



Obrázek 2.2 Zapojení vytvořené linky RS-485

Rozhraní SPI3 je použito pro komunikaci s externí FLASH pamětí AT45DB161. Paměť má velikost 16MB. Na toto rozhraní je připojena pouze paměť. Další dvě SPI rozhraní jsou použity pro komunikaci s okolními zařízeními. První rozhraní SPI je vyvedeno přímo na konektory, ke kterým se připojí pomocí vodičů řídicí jednotka zařízení stanice dopravníku. Druhé rozhraní SPI je použito pouze u větších displejů (128x64 a 240x64 bodů). Protože displeje budou použity v jiskrově bezpečných systémech APD, kde displej mohou ovládat dvě různé řídicí jednotky, je zapojen obvod

ADuM3151. Tento speciální obvod od firmy Analog Device zajistí galvanické oddělení pro SPI rozhraní.

V případě poklesu napájecího napětí je na resetovací pin mikroprocesoru zapojen obvod DS1818. Ten zajistí korektní reset mikroprocesoru při změnách napájecího napětí. Při poklesu pod 3,3V je na NRST pinu log. úroveň L. Při opětovném napětí 3,3V (tolerance 5%) bude na NRST pinu log. úroveň H. Toto řešení zajistí, aby nedocházelo k nedefinovaným stavům na jednotlivých pinech mikroprocesoru.

Na výstupní port C mikroprocesoru je zapojen grafický displej. Protože displej používá 5V logiku (mikroprocesor pouze 3,3V) je nutné zapojit převodník logických úrovní. Pro datové piny displeje je použitý osmi kanálový převodník SN74LVC8T245 od Texas Instruments. Displej s rozlišením 192x32 využívá pouze tři řídicí piny a proto by zapojení osmi kanálového převodníku bylo zbytečné. Je použit pouze čtyř kanálový převodník NTS0104 od NXP Semiconductors. U větších displejů je pro řízení využito šest pinů, proto je opět použit již zmíněný převodník SN74LVC8T245. Na port B (PB0 – PB7) jsou zapojeny datové piny, pomocí kterých se posílají data na původní alfanumerický displej. Řídicí piny alfanumerického displeje jsou zapojeny na portu A (PA0 – PA2). Přestože jsou úrovně vstupního signálu 0 – 5V a mikroprocesor má 3,3V logiku, nejsou zapojeny převodníky logických úrovní. Vstupní piny jsou tolerantní k 5V logice.

Kontrast displeje se nastavuje trimrem, který je zapojen mezi +5V a -5V. Některé displeje si záporné napětí vytváří sami, např. zvolené displeje s rozlišením 192x32 a 240x64 bodů. U displeje 128x64 bodů záporné napětí není vytvářeno přímo na displeji, proto musí být vytvořeno externě. K tomu slouží obvod ICL7660, který ze vstupního napětí vytvoří napětí s opačnou polaritou a stejnou úrovní. Tento obvod pro svoji funkčnost potřebuje externě připojit dva kondenzátory. Podle doporučení výrobce v datasheetu [16] jsou zapojeny kondenzátory s kapacitou 10μF.

Dále je zapojeno několik signalizačních LED diod. Jedna dioda signalizuje, zda je přivedeno napájecí napětí. Další tři diody jsou umístěny u mikroprocesoru, které slouží k signalizaci při běhu řídicího programu. Proud diodami je nastaven na 1mA pomocí předřadného odporu s hodnotou 1kΩ. Přestože jsou použity LED diody různých barev (tzn. mají různý úbytek napětí v propustném směru), předřadný odpor je u všech stejný. To bude mít vliv na procházející proud diodou resp. svítivost. Výpočet odporu:

$$R_2 = \frac{U_{log. H} - U_{LED}}{I_{LED}} = \frac{3,3V - 2,3V}{1mA} = 1k\Omega \quad (2.5)$$

kde:  $U_{log. H}$  je napětí na výstupu mikroprocesoru při logické úrovni H,  $U_{LED}$  je úbytek napětí na LED diodě v propustném směru,  $I_{LED}$  proud procházející diodou.

## 2.2 Návrh DPS

Byly navrženy tři odlišné desky plošných spojů. Rozměry vycházely z rozměrů použitého displeje. Všechny navržené DPS jsou dvouvrstvé.

Použité součástky kromě konektorů a odrušovacího filtru jsou v SMD provedení. Konektory pro připojení displeje jsou ve vrstvě TOP. Všechny ostatní součástky jsou ve vrstvě BOTTOM.

První navržená DPS je pro grafický displej s rozlišením 192x32 bodů. Její rozměry jsou 116x37 milimetrů. Šestnácti pinový konektor pro připojení ke grafickému displeji je na kratší straně DPS vpravo (při pohledu na vrstvu TOP). Konektor je ve dvou řadách, v každé po osmi pinech. Na protější straně je konektor, pomocí něž se navržený interface připojí k řídicí jednotce systému APD. Tento konektor má stejnou konfiguraci jako předchozí zmíněný. Posledními konektory umístěnými na DPS jsou šroubovací konektory pro linku RS-485 a pro rozhraní SPI2. Ty jsou umístěny na delší straně nahoře.

Další DPS je pro grafický displej s rozlišením 128x64 bodů s rozměry 98x70mm. Rozměry navržené DPS odpovídají rozměrům displeje. Grafický displej je připojen pomocí 20-pinového jednořadého konektoru, který je umístěn na dolním okraji DPS (při pohledu na vrstvu TOP). Na protější straně je 16-pinový konektor pro připojení k řídicí jednotce APD. Vedle něj je šroubovací konektor pro rozhraní SPI. Na kratší straně DPS vpravo jsou také šroubovací konektory, pomocí nichž se připojí linka RS-485 a galvanicky oddělené rozhraní SPI.

Poslední navržená DPS je pro největší displej 240x64 bodů. Její rozměry 98x65 mm jsou oproti rozměrům displeje menší. Důvodem je úspora materiálu na výrobu a samozřejmě cena DPS. Konektor pro grafický displej je umístěn na levé straně (při pohledu na vrstvu TOP). Jedná se o 20-pinový konektor v dvouřadé konfiguraci. Na protější straně je 16-pinový konektor pro připojení k řídicí jednotce APD. Šroubovací konektor pro galvanicky oddělené SPI je na dolním okraji DPS. Další šroubovací konektory pro linku RS-485 a další rozhraní SPI jsou na horním okraji.

## 3 NÁVRH SOFTWARE

Pro tvorbu obslužného programu byly použity volně dostupné vývojové nástroje od výrobce použitého mikroprocesoru. Jedná se o STM32CubeMX, kde se konfiguruje porty, dostupné periférie použité v mikroprocesoru, atd. V tomto programu byla vygenerována základní kostra programu. Následně byl kód editován v prostředí Atolic TrueStudio.

Při vytváření programu jsem využíval HAL knihovny.

### 3.1 Knihovny pro obsluhu displeje

Pro každý typ displeje byla vytvořena samostatná knihovna. Jejich názvy jsou odvozeny od použitého řadiče v grafickém displeji. Jsou to st7920.h, NT7108.h a RA6963.h

#### 3.1.1 St7920.h

Knihovna obsahuje základní funkce pro ovládání displeje, kterými jsou funkce pro zápis dat (Wdata) nebo instrukcí (Wcom) do displeje. Dále knihovna obsahuje funkce pro inicializaci, vymazání celého displeje, zápis znaků pro zobrazení, atd.

Zobrazitelná data se na grafickém displeji zapisují po jednotlivých řádcích. Displej je ve výchozím nastavení uzpůsoben k zobrazování znaků 8x16 (8 řádků x 16 sloupců). Tomu odpovídají jednotlivé adresy pozice znaku na displeji. Tzn. při použitém formátu 8x8 bodů na znak nelze znaky zapisovat do displeje po jednom. Proto se ve funkci pro zápis dat na displej vykresluje dva znaky najednou, viz funkce níže. Použitý formát znaků má u tohoto displeje rozlišení 8x8 bodů. Daný formát rozlišení obsahuje také mezery mezi jednotlivými řádky a znaky. Poslední řádek a sloupec v každém znaku je prázdný. Pro snížení paměťové náročnosti daného rozlišení je vytvořen font 7x8 bodů. Poslední osmý řádek, který je prázdný u všech znaků, je doplněn automaticky po zápisu předchozích řádků z paměti mikroprocesoru.

Funkce pro zápis znaků na grafickém displeji s řadičem ST7920.

```
void glcd_write_2char (char firstchar, char secondchar)
{
    firstchar -= 32;
    secondchar -= 32;

    if (Col <= 0x8F)
    {
        for (int j = 0; j < 7; j++)
        {
            Wcom (0x3E);
            Wcom (Row);
            Wcom (Col);
```

```

        Wdata(GLCD_ReadByteFromMemory((char*)(font7x8 + (7 * firstchar) + j)));
        Wdata(GLCD_ReadByteFromMemory((char*)(font7x8 + (7 * secondchar) + j)));
        Row++;
    }
    Wdata(0x00);
    Wdata(0x00);
    Col++;
    Row = Row - 7;
}
}
}

```

Tato funkce je využita pro zápis znaků přijatých pomocí paralelního rozhraní jako u displejů s řadičem HD44780. Při volání této funkce se předávají znaky k zobrazení ve formátu, ve kterém jsou definovány v ASCII tabulce (mezera má hodnotu 32). Proto na začátku funkce se od každého znaku odečítá hodnota 32. Následuje ověření globální proměnné Col, která definuje pozici znaku na displeji. Proměnná Col musí být nastavena ještě před voláním této funkce, jinak se zápis provádí za předchozími znaky. Pokud je hodnota nižší než 0x8F (tzn. znaky jsou v zobrazitelné části displeje) tak se provádí zápis dat na displej. V cyklu se zapisují jednotlivé řádky znaků. Nejdříve se odešle příkaz pro zápis, následovaný řádkem a sloupcem do kterých se zapisuje. Poté se odesílají samotná data, která jsou čtena z paměti mikroprocesoru. Proměnná Row znamená pozici řádku na displeji. Po skončení cyklu se zapisuje mezera mezi znaky (prázdný řádek, jak již bylo zmíněno). Následuje horizontální a vertikální posun k zápisu dalšího znaku.

Knihovna dále obsahuje funkci pro zápis znaků přijatých pomocí rozhraní SPI. Funkce se oproti předchozí liší v tom, že přijatá data neodpovídají znakům z ASCII tabulky. To znamená, že přijatá data přímo odpovídají znakům uloženým v paměti mikroprocesoru. Jedná se o jinou znakovou sadu než u displejů s řadičem HD44780. Znaková sada se nastavuje pomocí rozhraní UART (resp. linky RS-485), viz dále. Dostupné jsou dvě jazykové sady (vietnamština a arabština).

Funkce pro smazání displeje pouze zapisuje na celý displej prázdné řádky.

### 3.1.2 RA6963.h

Knihovna RA6963.h je podobná knihovně ST7920.h. Obsahuje funkce pro zápis dat (write\_data) a instrukcí (write\_command) do displeje, inicializaci displeje, smazání, zápis znaků přijatých pomocí SPI a paralelního rozhraní pro HD44780, ...

Zápis dat k zobrazení se provádí po řádcích jako v předchozím případě. Rozdíl spočívá v tom, že lze adresy jednotlivých znaků zapisovat samostatně. Nemusí se tedy zapisovat po dvou znacích.

### 3.1.3 NT7108.h

Funkce v této knihovně jsou obdobné jako v předcházejících knihovnách. Obsahuje funkce pro zápis dat (glcd\_writedata), zápis instrukcí (glcd\_writecommand), smazání displeje (glcd\_clearscreen), atd. Rozdíl spočívá hlavně v zápisu dat do displeje. Displej je rozdělen na dvě poloviny, které se ovládají samostatným řídicím signálem CS1 a CS2. Proto přibyly navíc dvě funkce. Jedna pro povolení práce s levou / pravou

(glcd\_enablecontroller) částí displeje a druhá pro zakázání práce (glcd\_disablecontroller). V těchto funkcích se pomocí přepínače switch vybere příslušná část displeje. Vstupním parametrem funkce je hodnota proměnné screen\_x (sloupec na displeji) dělená hodnotou 64 (počet sloupců ovládaných jedním řadičem).

```
switch(controller)
{
    case 0 :
        HAL_GPIO_WritePin(CS1_GRAPHIC_GPIO_Port,CS1_GRAPHIC_Pin,SET);
        break;
    case 1 :
        HAL_GPIO_WritePin(CS2_GRAPHIC_GPIO_Port,CS2_GRAPHIC_Pin,SET);
        break;
}
```

Zápis dat na displej je odlišný oproti předchozím displejům. Tento displej zapisuje data po sloupcích (předchozí zapisují data po řádcích). Aby displej mohl zobrazit co nejvíce znaků, je znakový font zredukován na 6x8 bodů. Znaky jsou opět odděleny prázdným řádkem a jednotlivé znaky odděluje prázdný sloupec. Jak již bylo zmíněno znaky se zapisují po sloupcích, proto poslední (šestý) sloupec není obsažen v paměti a zapisuje se automaticky po zvoleném znaku. Adresa sloupce, který následuje po aktuálně zapisovaném, se automaticky zvyšuje. Proto ve funkci zápisu znaku (viz níže) je pouze úprava ASCII kódu znaku a samotný zápis dat.

```
void glcd_writechar(char charToWrite)
{
    uint8_t i;
    charToWrite -= 32;
    for(i = 0; i < 5; i++)
        GLCD_WriteData(GLCD_ReadByteFromROMMemory((char*)(font5x8 +
            (5 * charToWrite) + i)));
    GLCD_WriteData(0x00);
}
```

Pro zápis znaků přijatých pomocí SPI knihovna obsahuje funkci glcd\_writecharspi. Funkce je obdobná jako pro zápis znaků přijatých pomocí paralelního rozhraní u HD44780. Jediný rozdíl spočívá ve znakové sadě, kterou lze na displeji zobrazit. Znaková sada pro znaky přijaté pomocí SPI se nastavuje pomocí UART (resp. linky RS-485), viz dále.

Před voláním této funkce je zapotřebí nastavit pozici, na kterou se budou nové znaky zapisovat. K tomu slouží funkce glcd\_goto, která obsahuje pouze řídicí příkazy

pro displej. Předlohou pro tuto knihovnu byla volně dostupná knihovna pro řadič KS0108 [17].

## 3.2 Knihovna pro práci s externí pamětí flash (flash\_spi.h)

V této knihovně jsou funkce pro práci s pamětí flash AT45DB161, která je připojena na rozhraní SPI3. V knihovně jsou funkce pro inicializaci paměti (init\_flash), zápis zvolené znakové sady (writeflash\_lang), čtení (readflash a readflash\_lang) a smazání celé paměti (clearflash). Poslední funkcí je flashready, která zjistí, zda je paměť připravena pro zápis nebo čtení.

Pro komunikaci s pamětí se musí odeslat definovaná posloupnost instrukcí. Ta vždy začíná tzv. opcode, což odpovídá příslušné instrukci v paměti. Následují tři adresní bajty a definovaný počet prázdných bajtů.

Funkce init\_flash slouží k inicializaci paměti. Zjišťují se data o použité paměti např. ID výrobce. Data jsou následně odeslána pomocí UART resp. linky RS-485 obsluze.

Zápis dat do paměti se provádí pouze při změně jazykové sady. Zapisuje se jeden bajt na pozici (adresu) 0x005 paměťového bloku. Tomu je uzpůsobena i funkce writeflash\_lang pro zápis do paměti, kde se funkci předává pouze hodnota, která se má uložit do paměti. Adresa paměťového bloku je pevně nastavena.

Tato knihovna je uzpůsobena pouze pro čtení dat z paměti, která se zobrazují na displeji. Tzn. nové znaky se musí být do paměti uloženy jiným obslužným programem.

Dále obsahuje funkce pro čtení. Jedná se o readflash\_lang, která se volá v hlavním kódu pouze po zapnutí zařízení. Funkce čte z paměťového místa 0x005 (kde zapisuje funkce writeflash\_lang) jeden bajt. Opět adresa paměťového bloku je pevně nastavena.

Pro čtení dat, která odpovídají jednotlivým zobrazitelným znakům na displeji, slouží funkce readflash. Vstupním parametrem této funkce je adresa paměťového bloku ze kterého se budou číst data. Přijatá data z paměti jsou uložena do globální proměnné rmsg. Jedná se o pole velikosti 70 bajtů pro font s rozlišením 7x8 znaků a o velikosti 50 bajtů pro font 5x8.

Pokud během práce s pamětí (zápis nebo čtení dat) došlo k chybě je pomocí UART, resp. linky RS-485, odeslána zpráva obsluze.

Vzorem pro vytvoření této knihovny byly firemní knihovny Hansen Electric spol. s r.o. Opava.

## 3.3 Knihovny pro změnu jazykové sady

### 3.3.1 ST7920\_spi.h

Tato knihovna slouží pro načtení jazykové sady z externí flash paměti. Obsahuje dvě funkce vietnamlang a arabiclang. V každé funkci se z definovaného místa v externí paměti flash přečtou data a uloží do globální proměnné fontspi. Jedná se o pole znaků a jeho velikost odpovídá největšímu počtu znaků v jazykových sadách. V tomto konkrétním případě má velikost 763 bajtů, což odpovídá 109 znakům vietnamské



jazykové sady (109 znaků \* 7 řádků na znak). O uložení dat na správnou pozici ve fontspi se stará cyklus for po každém přijetí dat. Dochází pouze ke kopírování dat z proměnné rmsg velikosti 70 bajtů do fontspi. V každém následujícím cyklu je pozice ve fontspi posunuta na konec předchozích dat. Po přečtení všech potřebných dat je uložena do paměti informace o použité jazykové sadě (funkce writeflash\_lang). Hodnota 0x01 odpovídá vietnamštině a hodnota 0x02 arabštině. Poté následuje odeslání zprávy po lince RS-485 o změně jazykové sady.

### 3.3.2 RA6963\_spi.h

Knihovna je stejná jako předchozí.

### 3.3.3 NT7108\_spi.h

Rozdíl oproti předchozím knihovnám spočívá v použitém fontu (jak již bylo zmíněno, ostatní displeje používají font 8x8, displej s radičem NT7108 používá font 6x8). Tomu odpovídá i menší paměťová náročnost na proměnnou fontspi. Její velikost je 545 bajtů (109 znaků \* 5 řádků). Data jsou uložena na stejných paměťových adresách jako v předchozích případech, ale jejich velikost je pouze 50 bajtů. Tomu odpovídá i globální proměnná rmsg, jejíž velikost se zmenšila ze 70 na 50 bajtů. Ukládání dat do fontspi z rmsg opět probíhá v cyklu. Po přečtení celé znakové sady se do paměti uloží záznam o jazyku a přes linku RS-485 se odešle zpráva o změně.

## 3.4 Knihovny pro radič HD44780

### 3.4.1 HD44780\_P1.h

Tato knihovna je vytvořena pro emulaci displeje s rozlišením 2x20 znaků. Obsahuje funkce pro zápis a čtení dat pomocí paralelního rozhraní jako u alfanumerických displejů s radičem HD44780.

Při zápisu dat (HD44780\_write\_data) nebo instrukcí (HD44780\_instruction) jsou vstupním parametrem osmi bitová data, která jsou přečtena z portu.

Ve funkci pro dekódování přijaté instrukce nejsou všechny možné kódy instrukcí, které obsahuje radič HD44780. Je to upraveno pouze na instrukce, které se využívají ve firmě pro komunikaci s displejem. Například není možnost zobrazit kurzor nebo možnost jeho blikání. Funkce HD44780\_instruction obsahuje kódy pro nastavení pozice kurzoru (HD44780\_AC), povolení nebo zakázání automatického posuvu kurzoru, nastavení pozice kurzoru na začátek displeje (vlevo nahoře) a vymazání všech dat z displeje. Součástí příkazu pro vymazání displeje je i naplnění pole string1 a string2 mezerami. Každé pole obsahuje maximálně 20 znaků, což odpovídá původnímu displeji, který měl rozlišení 2x20 znaků. Jednotlivé kódy instrukcí jsou definovány v hlavičce souboru.

Funkce HD44780\_write\_data zapisuje data na grafický displej. Před samotným zápisem se znak, který má být vykreslen na displeji, uloží do pole znaků (string1 nebo string2). Pro rozhodování do kterého pole se znak zapíše slouží proměnná pro pozici kurzoru (HD44780\_AC). Pokud je pozice menší než 20 zapisuje se do string1, v opačném

případě se zapisuje do string2. Poté následuje zápis dat na displej. Jak již bylo zmíněno, znaky se na grafický displej zapisují po dvojicích. Tomu odpovídají následující řádky kódu.

```
if(HD44780_AC % 2 == 0)
{
    if(HD44780_AC >= 0x40)
    {
        line2
        Col = char_position(HD44780_AC - 0x40);
        glcd_write_2char(string2[HD44780_AC - 0x40],string2[HD44780_AC - 0x3F]);
    }
    else
    {
        line1
        Col = char_position(HD44780_AC);
        glcd_write_2char(string1[HD44780_AC],string1[HD44780_AC + 1]);
    }
}
else
{
    if(HD44780_AC >= 0x40)
    {
        line2
        Col = char_position(HD44780_AC - 0x40);
        glcd_write_2char(string2[HD44780_AC - 0x41],string2[HD44780_AC - 0x40]);
    }
    else
    {
        line1
        Col = char_position(HD44780_AC);
        glcd_write_2char(string1[HD44780_AC - 1],string1[HD44780_AC]);
    }
}
```

První podmínka zjišťuje, zda se bude zapisovat lichý nebo sudý znak. V případě

sudého znaku následuje další podmínka. Ta rozhodne podle pozice kurzoru (HD44780\_AC), zda se bude zapisovat na první nebo druhý řádek. Makro line1 nebo line2 obsahuje adresu řádku na grafickém displeji. Jsou definovány v souboru st7920.h. Pozice znaku na řádku se zjistí pomocí funkce char\_position, kde vstupním parametrem je pozice kurzoru. Návratovou hodnotou z této funkce je adresa sloupce, na který se bude zapisovat. Následuje volání funkce pro zápis. Bližší popis této funkce je v odstavci 3.1.1. Rozdíl při zápisu sudého nebo lichého znaku je pouze při volání funkce glcd\_write2char. Při sudém znaku první parametr funkce je znak na pozici kurzoru a druhý parametr je znak následující za kurzorem. Při lichém znaku první parametr funkce je předchozí znak před kurzorem, druhý parametr funkce je znak, na který ukazuje kurzor. Po provedení předchozích řádků kódu následuje posun kurzoru (pokud je povolen). Pokud je kurzor na konci prvního řádku (má hodnotu 19) následuje zápis hodnoty 64. Touto hodnotou začíná v řadiči HD44780 druhý řádek. Pokud je kurzor na konci druhého řádku, neprovede se s hodnotou kurzoru nic. Pokud ani jedna z předchozích podmínek nebyla splněna, následuje zvýšení hodnoty kurzoru o 1.

Zápis znaků přijatých pomocí SPI (funkce spi\_write\_data) se provádí stejně. Jediný rozdíl je ve volané funkci pro zápis na displej (je volaná funkce glcd\_write\_2chaspi).

Funkce HD44780\_read\_data slouží pro čtení dat z displeje. Lze číst jak zapsaná data tak i pozici kurzoru. Při čtení kurzoru se v 8 bitu odesílá informace, zda je mikroprocesor zaneprázdněn (proměnná busyflag). Pokud je zaneprázdněn nemůže přijímat další data nebo instrukce. Po provedení všech požadovaných operací (mikroprocesor čeká na další povely) je hodnota v tomto bitu nulová. V opačném případě má hodnotu 1.

### 3.4.2 HD44780\_P2.h

Knihovna je podobná knihovně HD44780\_P1.h. Pomocí knihovny je emulována obsluha alfanumerického displeje s rozlišením 4x20 znaků. Proto jsou přidány navíc dvě pole string3 a string4. Do obou lze uložit maximálně 20 znaků.

Jsou vytvořeny funkce pro čtení a zápis dat nebo instrukcí.

Funkce pro zápis přijatého znaku na displej je rozdílná oproti předchozí. Zatím co v předchozím případě se znaky zapisovaly po dvou, v tomto případě se zapisuje po jednom znaku. To umožňuje zjednodušit obsluhu displeje. Na začátku funkce jsou podmínky, ve kterých se vyhodnocuje pozice kurzoru. Podle hodnoty se zapisuje do prvního řádku (hodnota od 0 do 19), druhého (64 - 83), třetího (20 - 39) a čtvrtého (84 - 103). Ukázka kódu pro zápis na první řádek:

```
if(HD44780_AC >= 0 && HD44780_AC < 0x14)
{
    glcd_goto(HD44780_AC * fontwidth, line1);
    glcd_writechar(data);
    string1[HD44780_AC] = data;
}
```

Příkazy uvnitř podmínky jsou podobné pro všechny čtyři řádky. Nejprve se nastaví adresa začátku zápisu dat na displej (funkce glcd\_goto). První parametr udává sloupce,

druhý parametr udává řádek. Zápis dat probíhá po sloupcích, proto je první parametr roven součinu kurzoru (HD44780\_AC) a použitého fontu (fontwidth). Fontwidth a line1 jsou makra. Následuje funkce pro zápis znaku na displej. Poté se znak, který se zapisoval na displej, uloží do pole string1. Následuje posuv kurzoru (pokud je povolen). Pokud se hodnota kurzoru nachází na konci řádku, uloží se hodnota první pozice následujícího řádku. Pokud je kurzor na čtvrtém řádku a poslední pozici, tak se hodnota kurzoru nezmění. Pokud žádná z podmínek nesouhlasí, provede se inkrementace hodnoty.

### 3.4.3 HD44780\_P3.h

Tato knihovna je vytvořena pro alfanumerický displej s rozlišením 4x40 znaků. Řízení tohoto displeje je odlišné od displejů 2x20 a 4x20 znaků. Obsahuje dva samostatné řídicí obvody. Každý obvod obsluhuje jednu polovinu displeje. Z tohoto důvodu jsou deklarované dvě proměnné pro kurzor. Ze stejného důvodu jsou i funkce definovány dvojité. Výjimku tvoří funkce pro zápis dat na displej. Ta je definována pouze jednou.

Na začátku této funkce je testovaná proměnná, která určuje komunikaci s horní nebo spodní polovinou displeje. Tato proměnná se nastavuje v hlavní smyčce programu. Pokud se pracuje s horní polovinou displeje, vykonávaný kód vypadá následovně:

```
if(HD44780_AC1 >= 0 && HD44780_AC1 < 0x1E)
{
    line1
    screen_x = HD44780_AC1 * GLCD_FONT_WIDTH;
    glcd_writechar(data);
    string1[HD44780_AC1] = data;
}
else if(HD44780_AC1 >= 0x40 && HD44780_AC1 < 0x5E)
{
    line2
    screen_x = (HD44780_AC1 - 0x40) * GLCD_FONT_WIDTH;
    glcd_writechar(data);
    string2[HD44780_AC1 - 0x40] = data;
}
```

Testuje se pozice kurzoru, zda se má zapisovat na první (rozsah 0 až 29) nebo druhý řádek (64 - 93). Počet znaků, které lze zobrazit je zredukován na 30 (oproti alfanumerickému displeji, kde lze zobrazit 40 znaků). To je dáno použitým fontem. Při zápisu na daný řádek se nejprve zvolí adresa řádku (definované makro např. line1). Následuje výpočet pozice znaku. Poté se volá funkce pro zápis dat na grafický displej. Dále se zobrazený znak uloží do pole. Po vykonání těchto instrukcí následuje posuv kurzoru (pokud je povolen).

Jak již bylo zmíněno, knihovna obsahuje dvě stejné funkce pro čtení dat z displeje.

Jediný rozdíl je v použité proměnné pro kurzor. To stejné platí i pro funkce pro dekodování instrukcí.

Posledními funkcemi v knihovně jsou funkce pro zápis znaků přijatých pomocí rozhraní SPI. U tohoto displeje jsou využity dvě rozhraní SPI. Výhodou je samostatné ovládání poloviny displeje jedním rozhraním.

## 3.5 Hlavní část programu

### 3.5.1 RGD1 P1

V hlavní části programu dochází k inicializaci periférií. Následuje inicializace flash paměti a přečtení informace o jazykové sadě. Následně podle přečteného údaje z paměti se rozhoduje, jaká znaková sada bude zkopírována z flash paměti do paměti mikroprocesoru. Implementovány jsou pouze dvě jazykové sady. Jedná se o vietnamštinu a arabštinu. Po vykonání těchto příkazů následuje inicializace UARTu. Tato periférie je použita s DMA v kruhovém režimu. To znamená, že během čtení dat se můžou provádět jiné instrukce. Následuje nastavení linky RS-485. Pro vysílání dat musí být na příslušném pinu obvodu MAX485 logická úroveň 1, pro příjem logická 0. Linka RS-485 je nastavena pro příjem dat, pouze během vysílání dojde k přepnutí směru dat. Po nastavení komunikační linky následuje inicializace grafického displeje.

V hlavní smyčce programu se zjišťuje, zda byly přijatá data přes UART. Zpracovávají se jednotlivé bajty v přijímacím bufferu. Následuje čtení pinu přiřazenému k SPI. Pokud je nulový je povolena komunikace pomocí tohoto rozhraní. Následuje testování proměnných, které značí přijatá data. `Newcharspi2` pro přijaté znaky a `newinstrspi2` pro přijatou instrukci. Tyto proměnné jsou nastavovány v obsluze přerušení SPI2. Při obsluze těchto proměnných je volána funkce pro zápis dat přijatých pomocí SPI. Vstupním parametrem funkce je druhý bajt pole `spi2rx`. První bajt určuje, zda se jedná o data nebo instrukce. Tento bajt je vyhodnocován v přerušení po příjmu dat od SPI. Po zápisu dat následuje vynulování proměnné `newcharspi2`, resp. `newinstrspi2`. Dále je pomocí rozhraní SPI, pomocí kterého byla přijata data, odeslána zpráva o provedení požadované operace. Odeslán je textový řetězec „OK“. Pro přijatá data je zápis kódu následující:

```
if(newcharspi2 == TRUE)
{
    spi_write_data(spi2rx[1]);
    newcharspi2 = FALSE;
    HAL_SPI_Transmit(&hspi2,spi2tx,2,1000);
}
```

Dalšími testovanými proměnnými jsou `newchar` a `newinstr`. Tyto proměnné jsou přiřazeny k obsluze řadiče HD44780. Nastavují se v přerušení od signálu Enable.

```

if(newchar == TRUE)
{
    HD44780_write_data(indata);
    newchar = FALSE;
    busyflag = 0;
}

```

Uvedený kód zapisuje data na grafický displej. Nejdříve je volaná funkce `HD44780_write_data`, kde vstupním parametrem je globální proměnná `indata`. Tato proměnná odpovídá přečteným datům z portu, v obsluze přerušení od signálu `Enable`. Následuje nulování proměnné značící nový přijatý znak. Posledním krokem je nulování proměnné `busyflag`, tzn. program čeká na další data.

Kruhový buffer pro UART má velikost 64 bajtů. Pokud se čtecí ukazatel nerovná zapisovacímu následuje přečtení dat z bufferu `uart_rx_buf`. Následuje inkrementace čtecího pointeru. Přečtená data z bufferu jsou vstupním parametrem funkce `uart_byte_available`. Použitý buffer je převzat z [18].

Funkce `uart_byte_available` ukládá přečtený tisknutelný znak (tzn. rozsah v ACCI tabulce 32 až 126) z `uart_rx_buf` do dalšího pomocného bufferu. Pokud narazí na znaky `\n` nebo `\r` ukončí se řetězec znaků v pomocném byffer znakem `NULL`. Následně se vyhodnocuje podmínka, zda jako první znak byla přijata vlnovka (tilda). Pokud podmínka platí, zavolá se funkce `uart_process_control`. Jejím vstupním parametrem je zmíněný pomocný buffer. V opačném případě se pomocné pole naplní nulami.

Funkce `uart_process_control` prochází vstupní data, až narazí v řetězci znaků na mezeru. Následuje vyhodnocení přijatého řetězce. Pomocí linky RS-485 lze odeslat tři různé řetězce. Jsou to `~help`, `~vietnam` a `~arabic`. Řetězec `~help` vypíše na linku RS-485 dostupné příkazy pro změnu jazyka. Řetězce `~vietnam` a `~arabic` zavolají funkce pro načtení znaků z paměti flash do mikroprocesoru.

Mikroprocesor má nastavené přerušení na pinu 2 portu A. Jedná se o signál `Enable` pro řadič `HD44780`. Přerušení je vyvoláno při vzestupné hraně signálu. V obsluze přerušení následují tři prázdné instrukce a čtení pinů přiřazených pro signály `RS` a `RW`. Poté je zkontrolováno, zda je signál `Enable` stále v logické 1. Tato posloupnost instrukcí zajistí správné časové reagování na čtení dat. Pokud by přerušení bylo vyvoláno nějakým krátkodobým rušením, tak data z portu nebudou přečtena a dále vyhodnocována. Pokud se nejednalo o falešnou změnu stavu signálu `Enable`, pokračuje se v obsluze přerušení. Rozhodne se podle přečtené úrovně signálu `RW`, zda se mají data do mikroprocesoru zapisovat nebo se má z něj číst. Jedná-li se o čtení dat na výstupní port `B` se zapíše data. Data k zápisu jsou návratovou hodnotou z funkce `HD44780_read_data` (funkce je blíže popsána v kapitole [3.4.1](#)). Následuje čekací smyčka, která je vytvořena pomocí `while`. Čeká se na sestupnou hranu signálu `Enable`. V případě čtení dat jsou relevantní data dostupná po sestupné hraně signálu. Následuje vyhodnocení stavu na pinu `RS` (čtou se data nebo instrukce). V příslušné části kódu dojde k přečtení hodnoty z portu `B` a nastavení proměnné značící přijatá data. Proměnná `newchar` značí nový přijatý znak, proměnná `newinstr` znamená přijatou instrukci. Během obsluhy tohoto přerušení je mikroprocesoru zakázáno opětovné přerušení od signálu `Enable`. V případě rušení

v tomto signálu by se obsluha přerušení mohla vyvolávat pořád dokola a došlo by ke zhroucení programu.

### 3.5.2 RGD1\_P2

Základní funkce programu je stejná jako v předchozím případě. Rozdíl spočívá v přidání rozhraní SPI1. Tomu odpovídá upravená část hlavní smyčky. Jsou doplněny proměnné, odpovídající přijatým datům pomocí rozhraní SPI1. Jedná se o `newcharspi1` a `newinstrspi1`.

### 3.5.3 RGD1\_P3

Posledním vytvořeným obslužným programem je obsluha grafického displeje s rozlišením 240x64 bodů. Jak již bylo zmíněno, alfanumerický displej s rozlišením 4x40 znaků obsluhují dva řadiče HD44780. Tomu odpovídají i vytvořené proměnné. Pro první polovinu displeje slouží proměnné `newchar1` a `newinstr1`. Druhé polovině displeje jsou určeny proměnné `newchar2` a `newinstr2`. Grafický displej také mohou obsluhovat dvě rozhraní SPI.

## 3.6 Obslužný program pro zápis dat do flash paměti

Flash paměť je určena pro uložení znakových sad pro grafický displej. Je uložena znaková sada pro zobrazení vietnamštiny a arabštiny.

Byly vytvořené dva obslužné programy z důvodu dvou znakových fontů (6x8 a 8x8).

Pro zápis fontu 8x8 (resp. 7x8 + mezera, která je přidána až při zápisu na displej) je vytvořena globální proměnná znaky o velikosti 70 bajtů. Na začátku programu dojde k inicializaci používaných periférií. Jedná se o rozhraní SPI3 a UART. Rozhraní SPI3 slouží pro komunikaci s pamětí a rozhraní UART komunikuje pomocí linky RS-485 s obsluhou zařízení. Následuje nastavení linky RS-485 pro vysílání. Dále se pokračuje v zápisu dat do paměti flash. Nejdříve se z paměti mikroprocesoru zkopíruje 10 znaků fontu pro vietnamštinu do proměnné znaky. Následuje inicializace flash paměti a zjištění, zda se mohou data zapisovat. Poté následuje samotný zápis. Adresa paměťového místa ve flash paměti je pevně stanovena. Pro vietnamštinu se zapisuje od adresy 0x011 po adresu 0x021. Arabština začíná na adrese 0x051 a končí na adrese 0x058. Po zápisu dat do paměti následuje čtení těchto zapsaných dat. Takto se pokračuje až po dosažení konce fontu. Počet bajtů ve fontu pro vietnamštinu je 763 (109 znaků). Protože se zapisuje po 70 bajtech (celkem se zapisuje 770 bajtů), jsou poslední zapisované bajty vyplněny nulami. Stejně se pokračuje i při zápisu arabštiny. Tento font obsahuje méně znaků (74) oproti vietnamštině. Po zápisu a čtení z posledního paměťového místa je pomocí UART odeslána zpráva obsluze „Flash programmed“. Tento program je určen pro grafické displeje s rozlišením 192x32 a 240x64 bodů.

Aby byla zachována stejné využití paměťových míst pro font 6x8 opět se zapisuje po 10 znacích. Globální proměnná znaky byla snížena na velikost 50 bajtů (10 znaků po 5 bajtech). Stejně jako u předchozího zápisu je konec fontu zapsaného do paměti doplněn nulami. Po provedení zápisu je opět vypsaná hláška „Flash programmed“. Tento obslužný program je určen pro grafický displej s rozlišením 128x64 bodů.

Funkce pro inicializaci, čtení flash paměti a zjištění, zda je připravena pro příjem dalších instrukcí, jsou stejné jako v knihovně `flash_spi.h`. V této knihovně byla implementována funkce pouze pro zápis zvolené jazykové sady. Byla pevně nastavena adresa. Funkce pro zápis dat v těchto obslužných programech je upravena. Úprava spočívá v přidání vstupního parametru funkce. Tímto parametrem se předává adresa paměťového místa ve flash paměti.



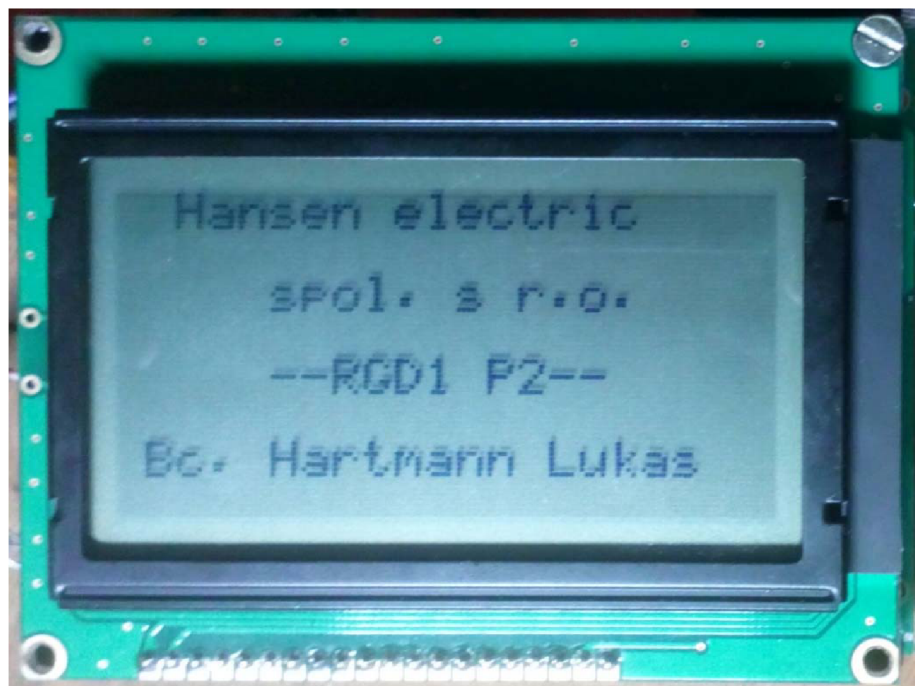
## 4 TESTOVÁNÍ NAVRŽENÉHO ZAŘÍZENÍ

Navržené zařízení bylo otestováno pomocí vytvořeného testovacího přípravku. Jednalo se o STM32VLDISCOVERY kit. Pro ověření funkčnosti komunikace byla nejprve otestována komunikace s alfanumerickým displejem. Po ověření funkční komunikace s displejem bylo následně připojeno navržené zařízení. Výsledek testování grafických displejů je uveden na následujících fotografiích. Data na displej jsou odesílána pomocí paralelního rozhraní jako u alfanumerických displejů s řadičem HD44780.

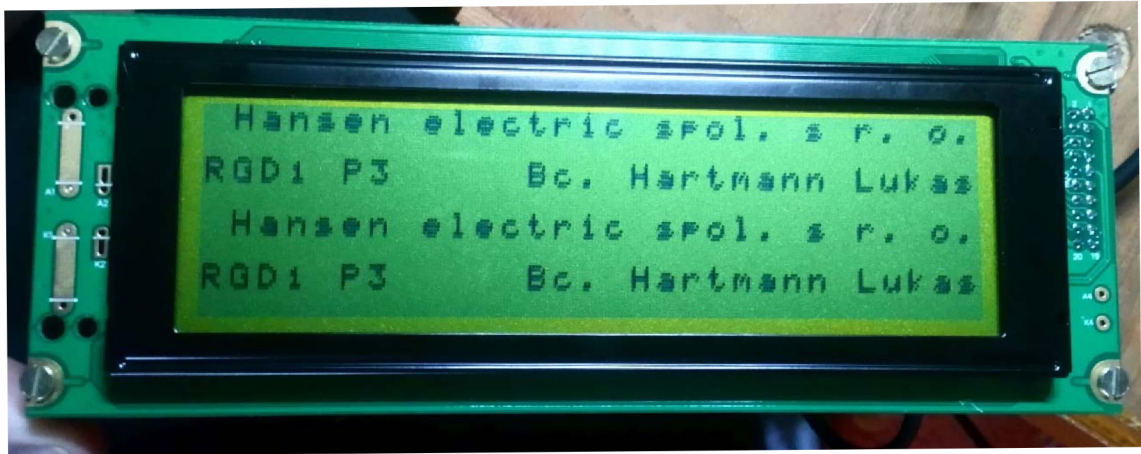
Při prvním testu zařízení jsem zjistil, že u displeje s rozlišením 192x32 bodů lze pozorovat jednotlivé zobrazitelné body na displeji. U zbylých dvou toto nebylo možné. Proto jsem změřil napětí na pinu nastavující kontrast u grafického displeje a tuto hodnotu porovnal s datasheetem. Napětí nebylo v potřebných mezích. Při zvýšení hodnoty odporu trimru na 20k $\Omega$  již nastavené napětí odpovídalo požadavkům v datasheetu.



Obrázek 4.1 Grafický displej 192x32 bodů



Obrázek 4.2 Grafický displej 128x64



Obrázek 4.3 Grafický displej 240x64 bodů

## 5 ZÁVĚR

V této práci bylo navrženo řešení pro náhradu alfanumerických displejů s řadičem HD44780 pomocí grafických displejů. Nahrazovány jsou alfanumerické displeje s rozlišením 2x20, 4x20 a 4x40 znaků. Použité grafické displeje mají rozlišení 192x32, 128x64 a 240x64 bodů. Pro každý displej byla navržena samostatná DPS, která bude umístěna pod displejem.

Znaky k zobrazení lze odesílat na displej dvěma způsoby. Prvním způsobem je komunikace pomocí paralelního rozhraní jako u alfanumerických displejů s řadičem HD44780. Druhou možností posílání dat na displej je pomocí rozhraní SPI. Grafický displej s rozlišením 192x32 bodů obsahuje pouze jedno rozhraní SPI. Ostatní displeje obsahují dvě rozhraní SPI, které jsou vzájemně galvanicky odděleny pomocí obvodu ADuM3151. Při komunikaci rozhraním SPI lze zobrazovat vietnamskou nebo arabskou jazykovou sadu. Výběr jazykové sady se provádí pomocí rozhraní UART resp. linky RS-485. Tyto jazykové sady jsou uloženy v externí paměti flash. Při změně jazykové sady dojde ke zkopírování dat z externí paměti do mikroprocesoru.

# LITERATURA

- [1] HARTMANN, L. *Přepínač hovorů pro systém APD*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 31 s., 19 s. příloh. Vedoucí bakalářské práce doc. Ing. Tomáš Frýza, Ph.D..
- [2] *DPS - plošné spoje od A do Z: odborný časopis pro vývoj a výrobu v oboru elektroniky*. Liberec: CADware, 2010-. ISSN 1804-4891.
- [3] ARM wikipedia.org. [online]. [cit 2018-12-10]. Dostupné z: <https://cs.wikipedia.org/wiki/ARM>
- [4] PAVLÍK, R. *Vývoj aplikací pro MCU/ARM*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2016. 25 s., 9 s. příloh. Bakalářská práce. Vedoucí práce: doc. Ing. Tomáš Frýza, Ph.D.
- [5] *Architektura mikrořadičů s jádrem ARM Cortex-M4* [online]. [cit. 2018-12-10]. Dostupné z: <https://www.root.cz/clanky/architektura-mikroradicu-s-jadry-arm-cortex-m4/>
- [6] CHYTIL, J. *Seriál - PIC16F84 - 25 - Sériový přenos*. [online]. 2008, [cit 2018-12-10]. Dostupné z: <http://www.8bitu.cz/clanek/serial-pic16f84-25-seriovy-prenos/>
- [7] FRÝZA T., *Řízení sériové komunikace: přednáška 7*. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky
- [8] Datasheet: MAX48x [online]. [cit 2018-12-10]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf>
- [9] *Externí sériové sběrnice SPI a I<sup>2</sup>C* [online]. [cit. 2018-12-10]. Dostupné z: <https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>
- [10] *Zobrazovací jednotky* [online]. [cit. 2018-12-10]. Dostupné z: [https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=9997](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=9997)
- [11] *Alfanumerické displeje* [online]. [cit. 2018-12-10]. Dostupné z: [http://home.zcu.cz/~dudacek/PZ/alfanum\\_displeje.pdf](http://home.zcu.cz/~dudacek/PZ/alfanum_displeje.pdf)
- [12] *Grafické LCD displeje a jejich řízení* [online]. [cit. 2018-12-10]. Dostupné z: <https://vyvoj.hw.cz/soucastky/graficke-lcd-displeje-a-jejich-řízení.html>
- [13] *Sitronix.com Chinese Fonts built in LCD controller/driver ST7920* [online]. Rev. 2008 [cit. 2019-05-16]. Dostupné z: <http://www.sitronix.com.tw/sitronix/product.nsf/Doc/ST7920>
- [14] *Neotec Semiconductor Ltd. LCD controller NT7108* [online]. Rev 2002 [cit. 2019-05-16]. Dostupné z: <https://www.crystalfontz.com/controllers/Neotec/NT7108>
- [15] *RAiO Technology Inc.. LCD controller RA6963* [online]. Rev 2008 [cit. 2019-05-16]. Dostupné z: [https://www.newhavendisplay.com/app\\_notes/RA6963.pdf](https://www.newhavendisplay.com/app_notes/RA6963.pdf)
- [16] *Renesas, ICL7660, ICL7660A* [online]. Rev. 7.00 [cit. 2019-05-16]. Dostupné z: <https://www.renesas.com/eu/en/www/doc/datasheet/icl7660.pdf>

- [17] *Universal C library for KS0108 based LCD displays* [online]. [cit. 2019-05-16]. Dostupné z: <http://en.radzio.dxp.pl/ks0108/>
- [18] Ing. Aleš Povalač, Ph.D., *MPOA cvičení č.10* [online]. 2016 [cit. 2019-05-11]. Dostupné z: [https://moodle.vutbr.cz/pluginfile.php/435085/mod\\_resource/content/2/MPOA\\_C10.pdf](https://moodle.vutbr.cz/pluginfile.php/435085/mod_resource/content/2/MPOA_C10.pdf)

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

APD	automatika pásových dopravníků
SPI	serial peripheral interface, sériové periferní rozhraní
USART	Universal synchronous/asynchronous receiver transmitter
I2C	Internal integrated circuit, interní datová sběrnice
RISC	Reduced Instruction Set Computing, redukováná instrukční sada
SMD	Surface Mounted Device, součástka pro povrchovou montáž
DPS	deska plošných spojů
DIP	Dual In-Line, dva v jedné řadě
MPU	Memory protection unit, jednotka pro ochranu paměti

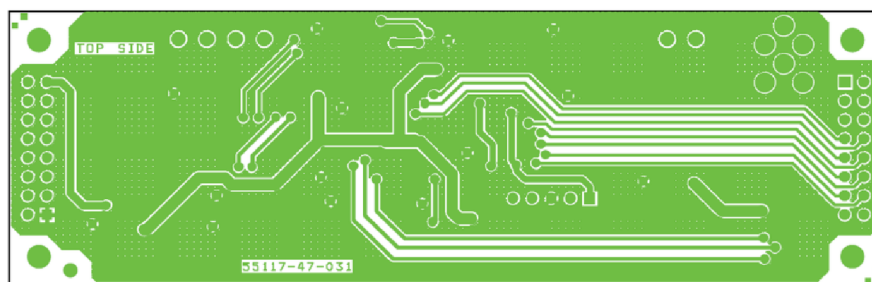
# PŘÍLOHY

A NÁVRH ZAŘÍZENÍ.....	32
<b>A.1 Displej s rozlišením 192x32 .....</b>	<b>32</b>
A.1.1 Deska plošného spoje - TOP .....	32
A.1.2 Deska plošného spoje - BOTTOM .....	32
A.1.3 Osazovací plán - TOP .....	32
A.1.4 Osazovací plán – BOTTOM .....	33
A.1.5 Seznam součástek .....	33
<b>A.2 Displej s rozlišením 128x64 .....</b>	<b>34</b>
A.2.1 Deska plošného spoje - TOP .....	34
A.2.2 Deska plošného spoje - BOTTOM .....	34
A.2.3 Osazovací plán - TOP .....	35
A.2.4 Osazovací plán – BOTTOM .....	35
A.2.5 Seznam součástek .....	35
<b>A.3 Displej s rozlišením 240x64 .....</b>	<b>37</b>
A.3.1 Deska plošného spoje - TOP .....	37
A.3.2 Deska plošného spoje - BOTTOM .....	37
A.3.3 Osazovací plán - TOP .....	38
A.3.4 Osazovací plán – BOTTOM .....	38
A.3.5 Seznam součástek .....	38
<b>B FOTODOKUMENTACE NAVRŽENÉHO ZAŘÍZENÍ .....</b>	<b>40</b>

# A NÁVRH ZAŘÍZENÍ

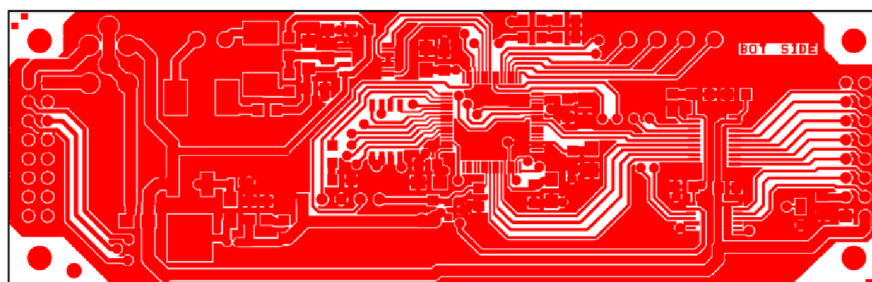
## A.1 Displej s rozlišením 192x32 bodů

### A.1.1 Deska plošného spoje – strana TO



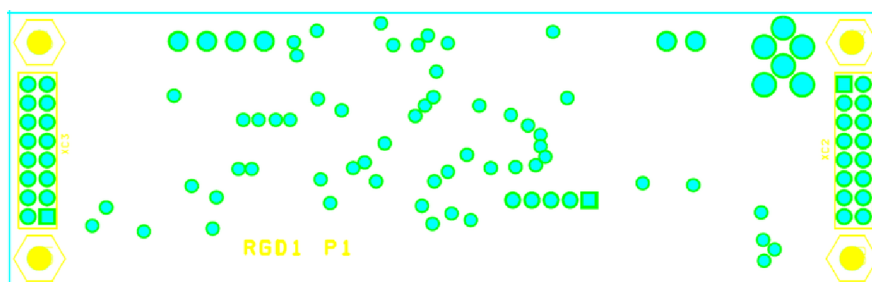
Rozměr desky 116 x 37 [mm], měřítko M1:1

### A.1.2 Deska plošného spoje – strana BOTTOM



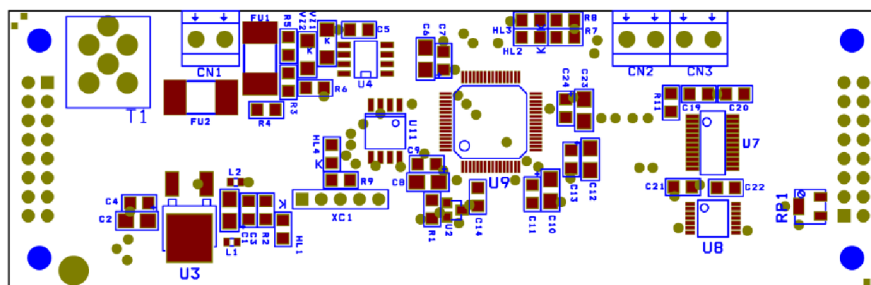
Rozměr desky 116 x 37 [mm], měřítko M1:1

### A.1.3 Osazovací plán – TOP





## A.1.4 Osazovací plán – BOTTOM



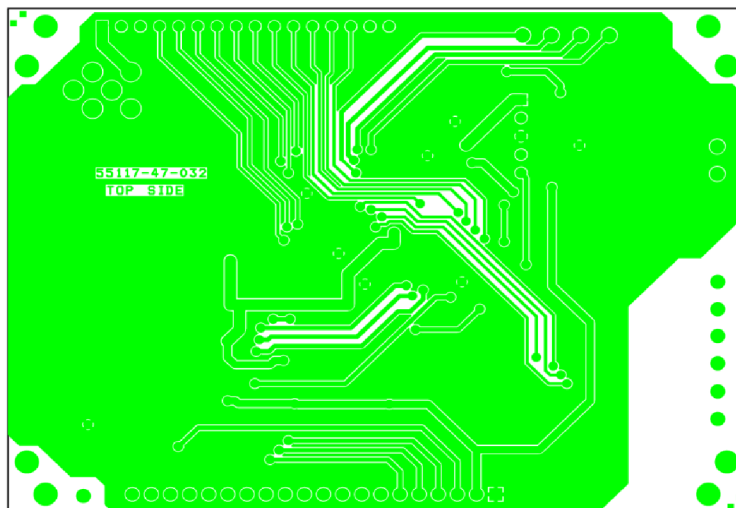
## A.1.5 Seznam součástek

Tabulka 1: Seznam součástek – displej 192x32

Označení	Hodnota	Pouzdro	Popis
CN1,CN2,CN3	MKDS 1/2-3,81		Konektor
C1,C2,C6,C8,C10,C12,C23	4U7/10V/A	A (1206)	Tantalový kondenzátor
C3,C4,C5,C7,C9,C11,C13,C14,C19,C20, C21,C22,C24	100nF	0805	Keramický kondenzátor
FU1,FU2	BSMD250-0,08	BSMD	Pojistka
HL1,HL2	Červená	0805	LED dioda
HL3	Zelená	0805	LED dioda
HL4	Žlutá	0805	LED dioda
L1,L2	BLM18KG1025	0603	Cívka
RP1	10k $\Omega$	3214	Trimr
R1,R5,R6	10k $\Omega$	0805	Rezistor
R2,R7,R8,R9	1k $\Omega$	0805	Rezistor
R3,R4	10 $\Omega$	0805	Rezistor
R11,R12	4k7 $\Omega$	0603	Rezistor
T1	BNX002-01	BNX002	Filtr
U2	DS1818	SOT23	Resetovací obvod
U3	MC33269	DPAK	Stabilizátor 3,3V
U4	MAX485	SOIC8	RS-485
U7,U8	SN74LVC8T245	SSOP24	Převodník napětí
U9	STM32L443	LQFP64	Mikroprocesor
U10	ADuM3151	SSOP20	Izolátor
U11	AT45DB161	SOIC8	Paměť
VZ1,VZ2	BZV55C5,6	SOD80C	Zenerova dioda
XC1	S1G2		Programovací konektor
XC3	S1G20		Konektor
XC4	S2G20		Konektor

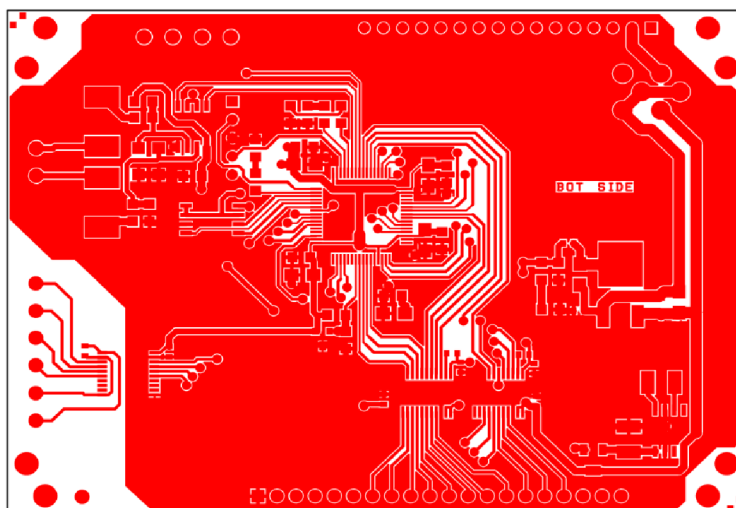
## A.2 Displej s rozlišením 128x64 bodů

### A.2.1 Deska plošného spoje – TOP



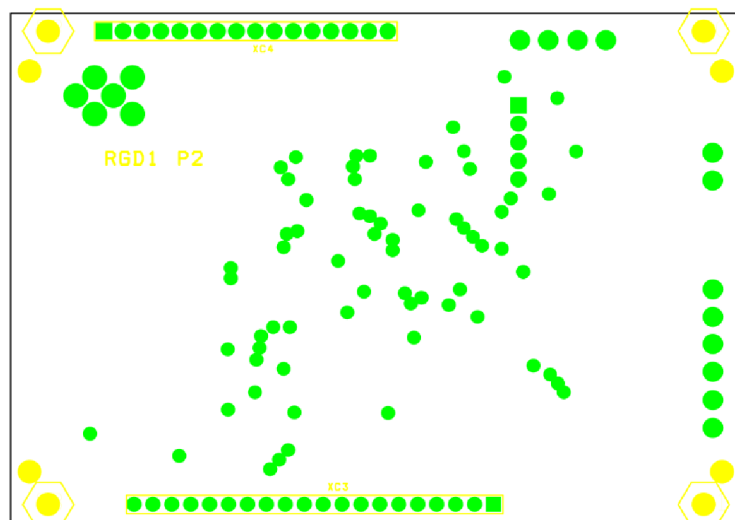
Rozměr desky 98 x 67 [mm], měřítko M1:1

### A.2.2 Deska plošného spoje – BOTTOM

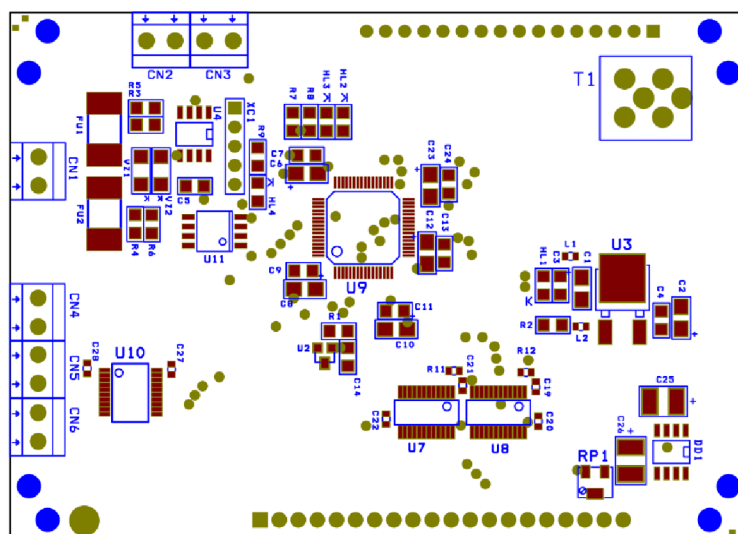


Rozměr desky 98 x 67 [mm], měřítko M1:1

### A.2.3 Osazovací plán – TOP



### A.2.4 Osazovací plán – BOTTOM



### A.2.5 Seznam součástek

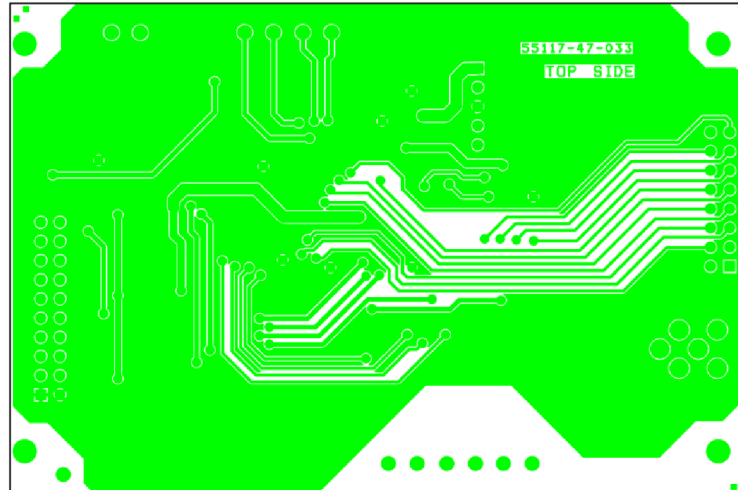
Tabulka 2: Seznam součástek – displej 128x64

Označení	Hodnota	Pouzdro	Popis
CN1,CN2,CN3,CN4,CN5,CN6	MKDS 1/2-3,81		Konektor
C1,C2,C6,C8,C10,C12,C23	4U7/10V/A	A (1206)	Tantalový kondenzátor
C3,C4,C5,C7,C9,C11,C13,C14,C24	100nF	0805	Keramický kondenzátor
C19,C20,C21,C22,C27,C28	100nF	0603	Keramický kondenzátor
C25,C26	10u/25V	B	Tantalový kondenzátor

DD1	ICL7660CBAZ	SOIC8	Nábojová pumpa
FU1,FU2	BSMD250-0,08	BSMD	Pojistka
HL1,HL2	Červená	0805	LED dioda
HL3	Zelená	0805	LED dioda
HL4	Žlutá	0805	LED dioda
L1,L2	BLM18KG1025	0603	Cívka
RP1	20k $\Omega$	3214	Trimr
R1,R5,R6	10k $\Omega$	0805	Rezistor
R2,R7,R8,R9	1k $\Omega$	0805	Rezistor
R3,R4	10 $\Omega$	0805	Rezistor
R11,R12	4k7 $\Omega$	0603	Rezistor
T1	BNX002-01	BNX002	Filtr
U2	DS1818	SOT23	Resetovací obvod
U3	MC33269	DPAK	Stabilizátor 3,3V
U4	MAX485	SOIC8	RS-485
U7,U8	SN74LVC8T245	SSOP24	Převodník napětí
U9	STM32L443	LQFP64	Mikroprocesor
U10	ADuM3151	SSOP20	Izolátor
U11	AT45DB161	SOIC8	Paměť
VZ1,VZ2	BZV55C5,6	SOD80C	Zenerova dioda
XC1	S1G2		Programovací konektor
XC3	S1G20		Konektor
XC4	S2G20		Konektor

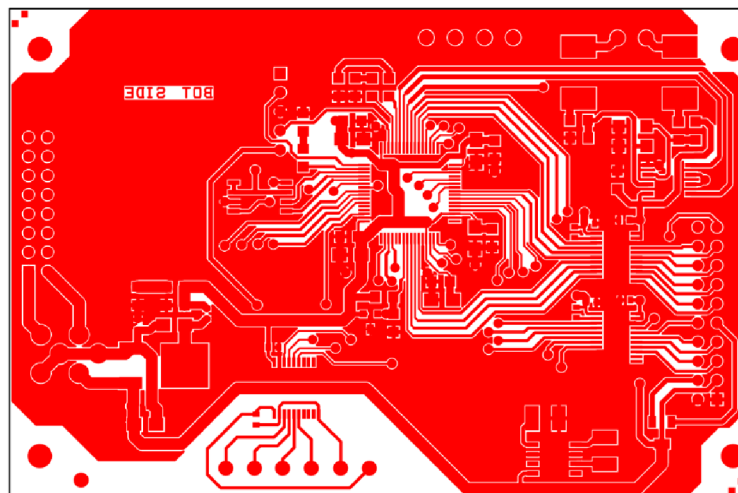
## A.3 Displej s rozlišením 240x68 bodů

### A.3.1 Deska plošného spoje – TOP



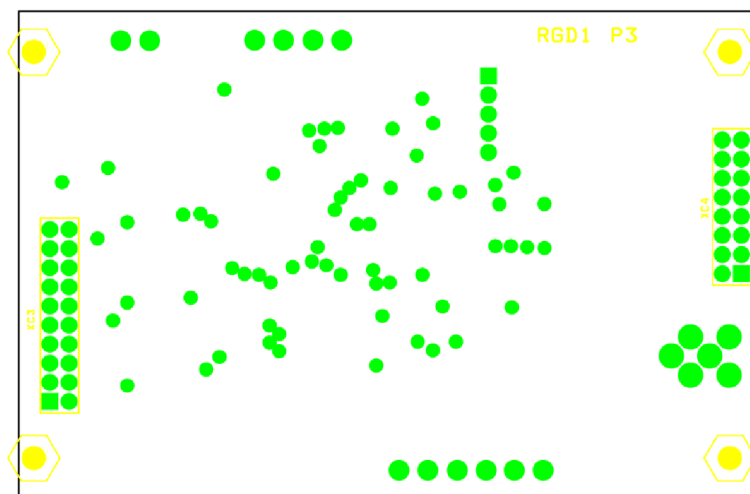
Rozměr desky 90 x 90 [mm], měřítko M1:1

### A.3.2 Deska plošného spoje – BOTTOM

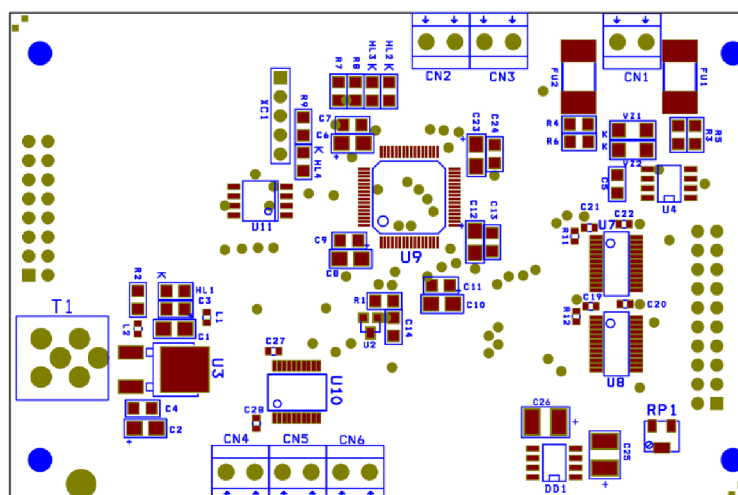


Rozměr desky 90 x 90 [mm], měřítko M1:1

### A.3.3 Osazovací plán – TOP



### A.3.4 Osazovací plán – BOTTOM



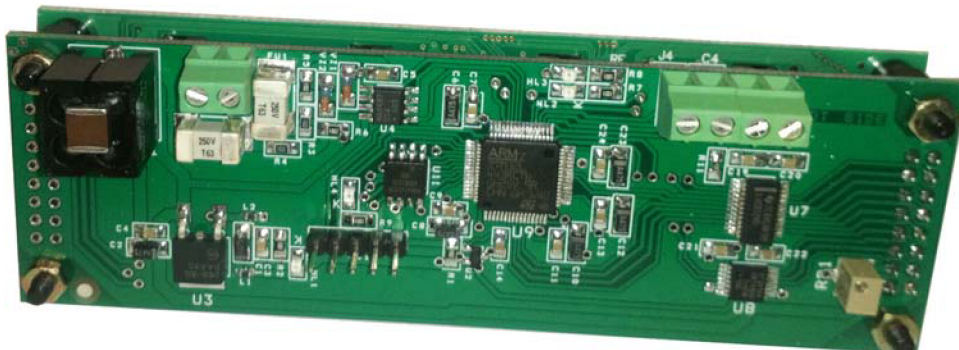
### A.3.5 Seznam součástek

Tabulka 3: Seznam součástek – displej 240x64

Označení	Hodnota	Pouzdro	Popis
CN1,CN2,CN3,CN4,CN5,CN6	MKDS 1/2-3,81		Konektor
C1,C2,C6,C8,C10,C12,C23	4U7/10V/A	A (1206)	Tantalový kondenzátor
C3,C4,C5,C7,C9,C11,C13,C14,C24	100nF	0805	Keramický kondenzátor
C19,C20,C21,C22,C27,C28	100nF	0603	Keramický kondenzátor
C25,C26	10u/25V	B	Tantalový kondenzátor
DD1	ICL7660CBAZ	SOIC8	Nábojová pumpa
FU1,FU2	BSMD250-0,08	BSMD	Pojistka
HL1,HL2	Červená	0805	LED dioda

HL3	Zelená	0805	LED dioda
HL4	Žlutá	0805	LED dioda
L1,L2	BLM18KG1025	0603	Cívka
RP1	20k $\Omega$	3214	Trimr
R1,R5,R6	10k $\Omega$	0805	Rezistor
R2,R7,R8,R9	1k $\Omega$	0805	Rezistor
R3,R4	10 $\Omega$	0805	Rezistor
R11,R12	4k7 $\Omega$	0603	Rezistor
T1	BNX002-01	BNX002	Filtr
U2	DS1818	SOT23	Resetovací obvod
U3	MC33269	DDPAK	Stabilizátor 3,3V
U4	MAX485	SOIC8	RS-485
U7,U8	SN74LVC8T245	SSOP24	Převodník napětí
U9	STM32L443	LQFP64	Mikroprocesor
U10	ADuM3151	SSOP20	Izolátor
U11	AT45DB161	SOIC8	Paměť
VZ1,VZ2	BZV55C5,6	SOD80C	Zenerova dioda
XC1	S1G2		Programovací konektor
XC3	S1G20		Konektor
XC4	S2G20		Konektor

## B FOTODOKUMENTACE NAVRŽENÉHO ZAŘÍZENÍ

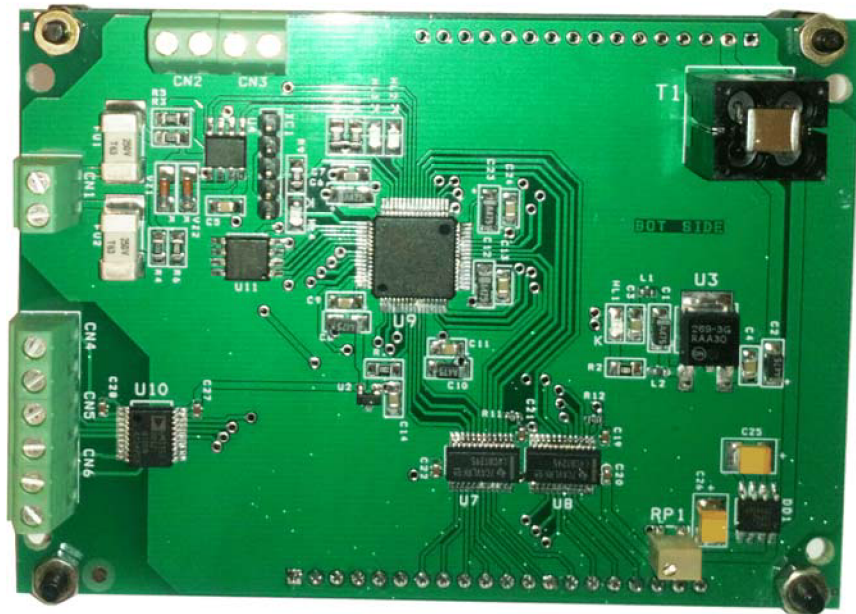


Obrázek B.1 Spodní strana displeje s rozlišením 192x32 bodů

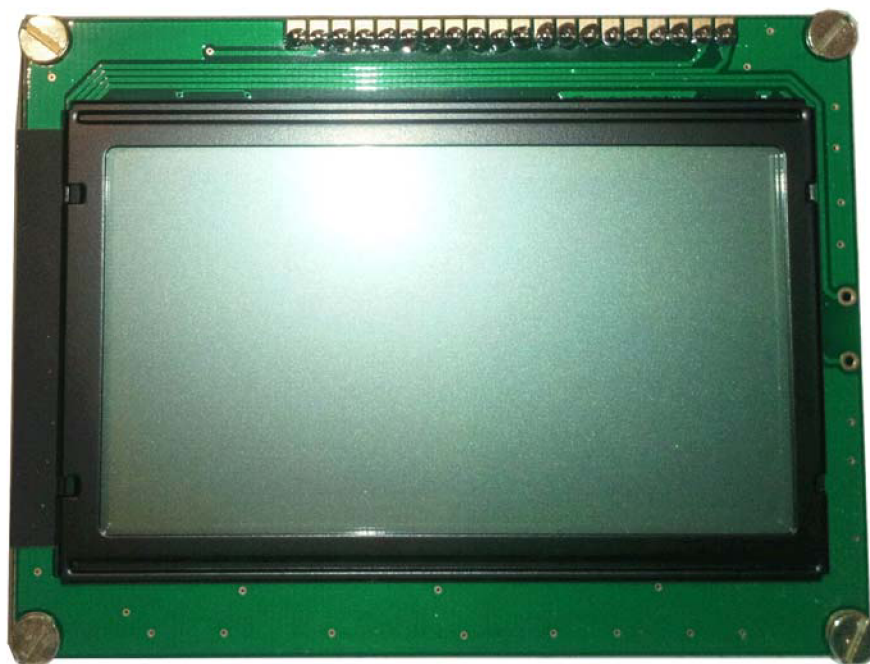


Obrázek B.2 Displej s rozlišením 192x32 bodů

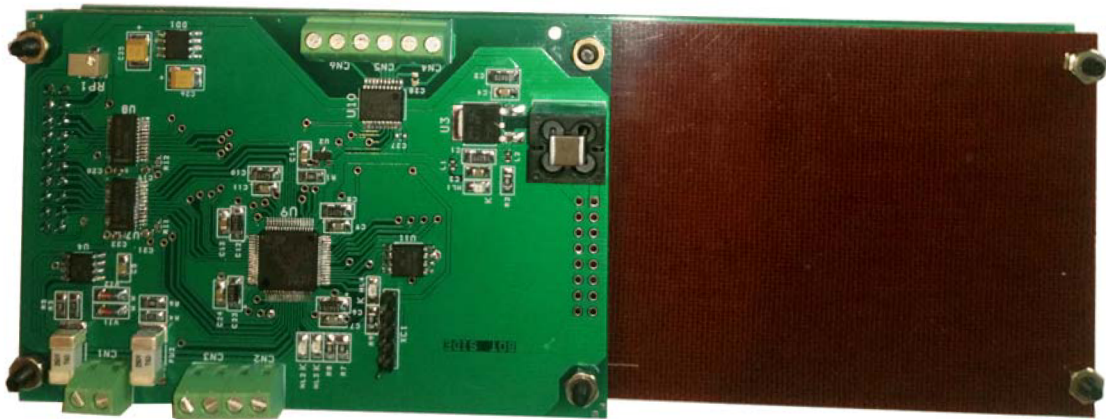




Obrázek B.3 Spodní strana displeje s rozlišením 128x64 bodů



Obrázek B.4 Displej s rozlišením 128x64 bodů



Obrázek B.5 Spodní strana displeje s rozlišením 240x64 bodů



Obrázek B.6 Displej s rozlišením 240x64 bodů