

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA ELEKTROTECHNICKÁ

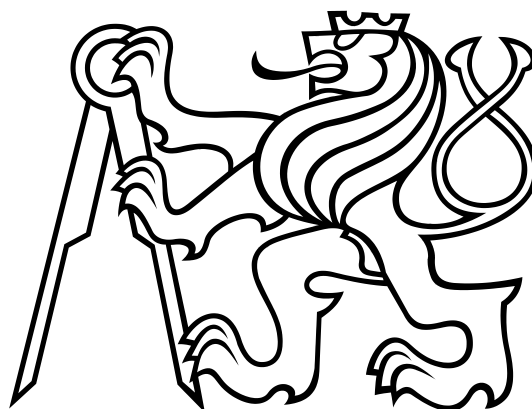
DIPLOMOVÁ PRÁCE

PRAHA 2015

**BC. MARTIN BORÝSEK**



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA ELEKTROTECHNICKÁ  
KYBERNETIKA A ROBOTIKA



DIPLOMOVÁ PRÁCE  
UNIVERZÁLNÍ INTELIGENTNÍ ŘADIČ PRO TFT LCD MODUL

PRAHA  
2014 – 2015

AUTOR: **BC. MARTIN BORÝSEK**  
VEDOUČÍ: **ING. RADEK SEDLÁČEK, PH.D.**



## PROHLÁŠENÍ

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne \_\_\_\_\_

\_\_\_\_\_

podpis



## PODĚKOVÁNÍ

Děkuji vedoucímu práce, panu Ing. Radku Sedláčkovi, Ph.D., za jeho čas strávený nad touto prací. Velmi si cením jeho ochoty a vstřícnosti, protože vzniklé nesnáze vždy řešil velmi trpělivě a přinesl k nim konstruktivní a věcné připomínky.

Důležité pro mě bylo zázemí poskytnuté klubem Silicon Hill, zejména při tvorbě a testování hardware. Za pomoc s 3D tiskem platformy pro uchycení displeje bych rád poděkoval Vojtěchu Sukovi.

Vážím si i opory rodiny v těžkých chvílích.





## ANOTACE

Cílem práce bylo navržení řadiče pro TFT LCD displeje, určeného pro vestavěné systémy. Důraz byl kladen na podporu libovolného panelu s dotykovou vrstvou, čehož bylo dosaženo navržením univerzálního konektoru a dvou možných způsobů připojení – jednoduché pasivní redukce pro konkrétní typ displeje a aktivní, programovatelné redukce s CPLD (Complex Programmable Logic Device).

Hlavní deska byla založena na mikrokontroléru STM32F439 s důrazem na maximální využití integrovaného grafického akcelerátoru DMA2D (Chrom-Art Accelerator). Aplikační paměť pro grafická primitiva a obrazový buffer jsou umístěny v externí paměti SDRAM. Desku doplňují komunikační rozhraní USB, čip FTDI a UART. Audio výstup je realizován kodekem a zesilovačem třídy D.

Práce obsahuje vytvořené knihovny pro obsluhu jednotlivých periférií a demo kódy, které ukazují použití knihoven a testují některé jejich funkce.

## ANNOTATION

The main goal of this thesis is the development of TFT LCD display driver employed in embedded systems. Great attention is given to establishing compatibility of various touch screen panels, which is achieved by using an universal connector with support for two possible connection methods – simple passive reduction for the particular type of display and active, programmable reduction with CPLD (Complex Programmable Logic Device).

The main board is based on the STM32F439 microcontroller with emphasis on maximizing the use of integrated graphics accelerator DMA2D (Chrom-Art Accelerator). Application memory for graphical primitives and frame buffer are positioned in the external SDRAM memory. Board implements communication interfaces, such as USB, FTDI and UART. Audio output is realized using codec and D class amplifier.

This thesis includes libraries for manipulation of individual peripherals and demo codes, which are created with emphasis to illustrate the use of libraries and their particular functions.





## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Martin Borýsek**

Studijní program: **Kybernetika a robotika**  
Obor: **Senzory a přístrojová technika**

Název tématu česky: **Univerzální inteligentní řadič pro TFT LCD modul**

Název tématu anglicky: **Versatile Intelligent Controller for TFT LCD Module**

### Pokyny pro vypracování:

Navrhněte a realizujte univerzální inteligentní řadič pro TFT LCD moduly určený pro vestavné systémy. Cílem práce je nalézt řešení, které umožní připojit libovolný TFT LCD modul s dotykovým panelem s rozlišením alespoň 800 x 600 bodů. Řešení založte na použití procesoru řady ARM Cortex-M4, typ STM32F429/F439, který přímo obsahuje periférii pro řízení TFT LCD modulu. Procesorové jádro doplňte o aplikační externí paměť SDRAM (64 MB či více), audiokodek, komunikační rozhraní (UART, USB), dva A/D vstupy, dva D/A výstupy, sadou několika digitálních vstupů a výstupů a slotem pro microSD kartu. V rámci firmwaru řadiče vytvořte knihovnu základních funkcí pro obsluhu všech integrovaných periférií řadiče včetně TFT LCD modulu. Mezi základní funkce pro ovládání TFT LCD modulu bude patřit vykreslování grafických primitiv a textů různým fontem.

### Seznam odborné literatury:

- [1] ST MICROELECTRONICS. *Ref. manuál k STM32F439xx* [online]. [cit. 2014-11-04]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/DM00077036.pdf>
- [2] YIU, Joseph: *The definitive guide to ARM CORTEX-M3 and CORTEX-M4 processors*. 3rd ed. Oxford: Elsevier, c2014, xxxv, 818 s. ISBN 978-0-12-408082-9.
- [3] ALTERA. *MAX V Device Handbook* [online]. 2011 [cit. 2014-11-04]. Dostupné z: [http://www.altera.com/literature/hb/max-v/max5\\_handbook.pdf](http://www.altera.com/literature/hb/max-v/max5_handbook.pdf)

Vedoucí diplomové práce: Ing. Radek Sedláček, Ph.D.

Datum zadání diplomové práce: 12. listopadu 2014

Platnost zadání do<sup>1</sup>: 31. srpna 2016

L. S.

Doc. Ing. Jan Holub, Ph.D.  
vedoucí katedry

Prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 12. 11. 2014

<sup>1</sup> Platnost zadání je omezena na dobu tří následujících semestrů.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>19</b>
1.1	Současné trendy a budoucí vývoj displejů . . . . .	20
<b>2</b>	<b>Dostupná řešení pro TFT displeje</b>	<b>21</b>
<b>3</b>	<b>Teoretická část</b>	<b>23</b>
3.1	Typy zobrazovacích technologií . . . . .	23
3.2	Kódování barev . . . . .	24
3.3	Chyby kódování . . . . .	25
3.4	Snímání dotyků . . . . .	26
3.5	Bresenhamův algoritmus . . . . .	27
3.5.1	Odvození . . . . .	27
3.6	Vnitřní struktura BMP obrázku . . . . .	29
3.7	Architektura mikrokontroléru . . . . .	30
<b>4</b>	<b>Hardware TFT LCD řadiče</b>	<b>31</b>
4.1	Blokové schéma . . . . .	32
4.2	Mikrokontrolér . . . . .	33
4.3	Časování mikrokontroléru . . . . .	34
4.4	Grafické jádro . . . . .	35
4.5	LTDC periferie . . . . .	36
4.6	Časování videosignálu . . . . .	37
4.7	Řadič dotykové vrstvy . . . . .	39
4.7.1	Komunikační protokol s řadičem . . . . .	40
4.8	Paměťová karta microSD . . . . .	41
4.8.1	Endianita . . . . .	41
4.8.2	Komunikace s kartou . . . . .	42
4.8.3	Organizace dat na kartě . . . . .	42
4.8.4	Test rychlosti karty . . . . .	43
4.9	Paměť pro obrazová data . . . . .	44
4.9.1	Vnitřní struktura SDRAM . . . . .	45
4.9.2	Princip čtení a časování . . . . .	46
4.9.3	Proces inicializace SDRAM . . . . .	47
4.9.4	Zapojení SDRAM . . . . .	48
4.9.5	Test rychlosti SDRAM . . . . .	50
4.10	Implementovaná komunikační rozhraní . . . . .	51
4.10.1	RS-232 . . . . .	51
4.10.2	FTDI . . . . .	52
4.10.3	USB On-The-Go Fullspeed . . . . .	53
4.11	Audio . . . . .	54
4.11.1	Audiokodek WM8731 . . . . .	55
4.11.2	Sběrnice I2S . . . . .	56
4.11.3	Příklad nastavení audiokodeku . . . . .	57
4.11.4	Zesilovač TPA2012 . . . . .	58
4.12	Napájení . . . . .	59
4.13	GPIO piny, LED a tlačítko . . . . .	60
<b>5</b>	<b>Výstup na displej</b>	<b>61</b>

5.1	Univerzální konektor pro TFT LCD displej . . . . .	62
5.2	Redukce pro displej ET0700G0DH6 . . . . .	64
5.3	Programovatelná redukce CPLD . . . . .	65
5.3.1	Konfigurace vstupů a výstupů . . . . .	66
5.3.2	Programování CPLD . . . . .	68
<b>6</b>	<b>Software</b>	<b>69</b>
6.1	Podpora fontů . . . . .	70
6.2	Board Support Package driver . . . . .	71
6.3	Demonstrační kódy pro BSP . . . . .	72
6.3.1	GUI Demo . . . . .	73
6.3.2	Test podpory SD karty a knihovny FatFS . . . . .	74
6.3.3	Test rychlosti SD karty . . . . .	75
6.3.4	Test rychlosti SDRAM + RAMdisk . . . . .	76
6.3.5	Demo dotykové vrstvy a kalibrace . . . . .	77
6.4	Dostupné grafické knihovny pro STM32F4xx . . . . .	78
6.4.1	Touch GFX . . . . .	78
6.4.2	STemWin . . . . .	79
<b>7</b>	<b>Platforma pro uchycení displeje</b>	<b>81</b>
<b>8</b>	<b>Závěr</b>	<b>83</b>
<b>9</b>	<b>Přílohy</b>	<b>89</b>
9.1	Rozložení pinů paměti SDRAM TSOP18 (54 pin) . . . . .	89
9.2	Platforma pro uchycení PCB k displeji . . . . .	90
9.3	Nákres použitého displeje ET0700G0DH6 . . . . .	91
9.4	Plošný spoj hlavní desky . . . . .	92
9.5	Plošný spoj CPLD . . . . .	93
9.6	Plošný spoj redukce . . . . .	94
9.7	Render hotového zařízení . . . . .	95
9.8	Fotografie hotového zařízení . . . . .	97
<b>10</b>	<b>Obsah příloženého CD</b>	<b>99</b>

## Seznam obrázků

Obrázek 1:	Primitivní LCD displeje . . . . .	19
Obrázek 2:	Současné trendy ve využití TFT displejů . . . . .	20
Obrázek 3:	Řadič uLCD-43 od firmy 4D Systems . . . . .	21
Obrázek 4:	Modul G400-S od firmy GHI . . . . .	21
Obrázek 5:	Různé zobrazovací technologie . . . . .	23
Obrázek 6:	Ikona prohlížeče Google Chrome . . . . .	25
Obrázek 7:	Ikona prohlížeče Mozilla Firefox . . . . .	25
Obrázek 8:	Princip fungování dotykové vrstvy . . . . .	26
Obrázek 9:	Snímání polohy rezistivního dotykového displeje . . . . .	26
Obrázek 10:	Rasterizace vektoru pomocí Bresenhamova algoritmu . . . . .	27
Obrázek 11:	Maticová sběrnice 32-bit multi-AHB . . . . .	30
Obrázek 12:	Organizace schémat . . . . .	31
Obrázek 13:	Blokové schéma . . . . .	32
Obrázek 14:	Mikrokontrolér STM32F439BIT6U . . . . .	33
Obrázek 15:	Časování mikrokontroléru . . . . .	34
Obrázek 16:	Architektura grafické části řady STM32F4x9 . . . . .	35
Obrázek 17:	Architektura periferie LTDC . . . . .	36
Obrázek 18:	Časování LTDC periferie . . . . .	36
Obrázek 19:	Časování displeje ET0700G0DH6 . . . . .	37
Obrázek 20:	Schéma řadiče dotykové vrstvy . . . . .	39
Obrázek 21:	Touch report protokol . . . . .	40
Obrázek 22:	Command protokol . . . . .	40
Obrázek 23:	Paměťová karta . . . . .	42
Obrázek 24:	Test rychlosti paměťové karty microSD . . . . .	43
Obrázek 25:	Vnitřní struktura SDRAM . . . . .	45
Obrázek 26:	Princip čtení z paměti . . . . .	46
Obrázek 27:	Mode Register paměti SDRAM . . . . .	47
Obrázek 28:	Paměť SDRAM . . . . .	48
Obrázek 29:	Test rychlosti přístupu do SDRAM blokujícím módem, DMA a DMA2D . . . . .	50
Obrázek 30:	Blokové schéma převodníku UART/RS232 . . . . .	51
Obrázek 31:	Převodník mezi UART a RS232 . . . . .	51
Obrázek 32:	Připojení k USB pomocí FTDI čipu . . . . .	52
Obrázek 33:	USB On-The-Go Fullspeed . . . . .	53
Obrázek 34:	PWM modulace 25 % . . . . .	54
Obrázek 35:	PCM modulace . . . . .	54
Obrázek 36:	Audiokodek WM8731 . . . . .	55
Obrázek 37:	Časování sběrnice I2S . . . . .	56
Obrázek 38:	Průběh signálu na sběrnici I2S . . . . .	57
Obrázek 39:	Zesilovač TPA2012 . . . . .	58
Obrázek 40:	Spínané zdroje pro napájení desky a USB OTG . . . . .	59
Obrázek 41:	Účinnosti měničů . . . . .	59
Obrázek 42:	Zapojení tlačítek a LED diod na desce řadiče . . . . .	60
Obrázek 43:	Připojení libovolného LCD displeje k hlavní desce . . . . .	61
Obrázek 44:	Konektor Omron XF2M-4015-1A . . . . .	62
Obrázek 45:	Univerzální konektor . . . . .	63
Obrázek 46:	Schéma redukce pro displej ET0700G0DH6 . . . . .	64

Obrázek 47:	Schéma programovatelné redukce s CPLD . . . . .	65
Obrázek 48:	Vnitřní struktura CPLD . . . . .	65
Obrázek 49:	Mapování vstupů a výstupů do CPLD . . . . .	66
Obrázek 50:	CPLD programovací rozhraní JTAG . . . . .	67
Obrázek 51:	CPLD oscilátor . . . . .	67
Obrázek 52:	Napájení jádra CPLD . . . . .	67
Obrázek 53:	Struktura entity CPLD . . . . .	68
Obrázek 54:	Mapa zdrojového kódu . . . . .	69
Obrázek 55:	Typografické úpravy textu . . . . .	70
Obrázek 56:	Ukázka fontu Consolas . . . . .	70
Obrázek 57:	Ukázka fontu Lucida Console . . . . .	70
Obrázek 58:	Screenshot obrazovky dema GUI . . . . .	73
Obrázek 59:	Screenshot obrazovky testu microSD . . . . .	74
Obrázek 60:	Screenshot obrazovky testu rychlosti microSD . . . . .	75
Obrázek 61:	Screenshot obrazovky testu rychlosti SDRAM . . . . .	76
Obrázek 62:	Průběh kalibrace dotykové vrstvy . . . . .	77
Obrázek 63:	Demo počasí vytvořené s knihovnou TouchGFX . . . . .	78
Obrázek 64:	Demo termostatu vytvořené s knihovnou TouchGFX . . . . .	78
Obrázek 65:	Ukázka uživatelského rozhraní vytvořeného se STemWin . . . . .	79
Obrázek 66:	Graf vykreslený knihovnou STemWin . . . . .	79
Obrázek 67:	Platforma pro uchycení desek k displeji . . . . .	81
Obrázek 68:	Rozložení pinů paměti SDRAM TSOPII 54 pin . . . . .	89
Obrázek 69:	Výkres platformy pro uchycení PCB k displeji . . . . .	90
Obrázek 70:	Rozměry displeje ET0700G0DH6 . . . . .	91
Obrázek 71:	Hlavní deska - horní vrstva . . . . .	92
Obrázek 72:	Hlavní deska - dolní vrstva . . . . .	92
Obrázek 73:	CPLD - horní vrstva . . . . .	93
Obrázek 74:	CPLD - dolní vrstva . . . . .	93
Obrázek 75:	Redukce pro TFT LCD ET0700G0DH6 . . . . .	94
Obrázek 76:	Render hlavní desky řadiče (horní strana) . . . . .	95
Obrázek 77:	Render redukce s CPLD (horní strana) . . . . .	95
Obrázek 78:	Render pasivní redukce (spodní strana) . . . . .	96
Obrázek 79:	Hlavní deska řadiče (spodní strana) . . . . .	97
Obrázek 80:	Redukce s CPLD (horní strana) . . . . .	97
Obrázek 81:	Pasivní redukce (spodní strana) . . . . .	98



## Seznam tabulek

Tabulka 1:	Podporované barevné prostory . . . . .	24
Tabulka 2:	Chybná interpretace barev zahazením horních dvou bitů . . . . .	25
Tabulka 3:	Vnitřní struktura BMP obrázku . . . . .	29
Tabulka 4:	Časování pro displeje 800 x 480 . . . . .	38
Tabulka 5:	Konfigurace LTDC periferie . . . . .	38
Tabulka 6:	Mapování vodičů pro touchscreen . . . . .	39
Tabulka 7:	Touch report protokol . . . . .	40
Tabulka 8:	Připojení SD pomocí SDIO . . . . .	42
Tabulka 9:	Připojení SD pomocí SPI . . . . .	42
Tabulka 10:	Test rychlosti paměťové karty microSD . . . . .	43
Tabulka 11:	Časování pro SDRAM periferie FMC . . . . .	47
Tabulka 12:	Mapování vodičů periferie SDRAM . . . . .	49
Tabulka 13:	Příkazy pro paměť SDRAM . . . . .	49
Tabulka 14:	Rychlost přístupu do SDRAM pomocí blokujícího módu a DMA . . . . .	50
Tabulka 15:	Rychlost zápisu do SDRAM pomocí DMA2D . . . . .	50
Tabulka 16:	Připojení čipu FTDI k mikrokontroléru . . . . .	52
Tabulka 17:	Mapování pinů USB OTG Fullspeed . . . . .	53
Tabulka 18:	Mapování vodičů I2C . . . . .	55
Tabulka 19:	Mapování vodičů I2S . . . . .	55
Tabulka 20:	Pojmenování vodičů na sběrnici I2S . . . . .	56
Tabulka 21:	Zesílení zesilovače TPA2012 . . . . .	58
Tabulka 22:	Mapování vodičů zesilovače . . . . .	58
Tabulka 23:	Mapování pinů GPIO . . . . .	60
Tabulka 24:	Mapování LED . . . . .	60
Tabulka 25:	Mapování tlačítek na desce . . . . .	60
Tabulka 26:	Mapování pinů periferie LTDC . . . . .	63



# 1 ÚVOD

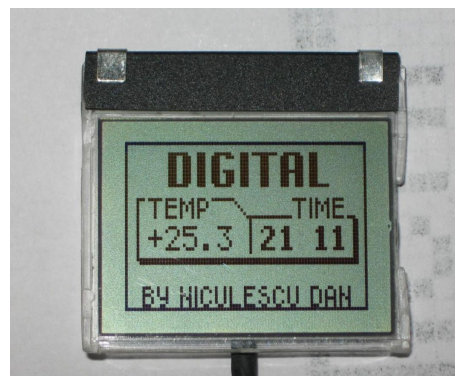
V dnešní době by byl svět bez displejů jen těžko představitelný. Jsou všude okolo nás a stále více zařízení je jimi vybaveno. Tím narůstá potřeba jejich integrací i do vestavěných zařízení a mezi samozřejmost už dneska považujeme i podporu dotykového rozhraní.

Dříve se zařízení ovládala pomocí tlačítek a přepínačů, jednotlivé stavy byly signalizovány jednoduchými kontrolkami. S postupem času a rozvojem LCD technologie se přešlo na používání primitivních, několikařádkových LCD displejů (obr. 1a). S přelomem tisíciletí přišel na trh mobilní telefon Nokia 3310, který používal monochromatický grafický displej s rozlišením 84x48 bodů (obr. 1b).

Obě zmíněné varianty obsahují vlastní integrovaný řadič. Používají se i dnes, zvláště u nenáročných aplikací ve spojení s malými a málo výkonnými mikrokontroléry (ATMEGA, PIC, Cortex M0). Komunikace probíhá buď po sběrnici SPI (popř. I2C), nebo několikabitovým paralelním rozhraním.



(a) Čtyřřádkový LCD displej



(b) Displej telefonu Nokia 3310

Obrázek 1: Primitivní LCD displeje

S rozvojem osobních asistentů a pokročilejších mobilních telefonů (PDA) a taktéž s příchodem první generace telefonu Apple iPhone byl rychle rozšířen standard dotykového ovládání a sofistikovanějšího uživatelského rozhraní. Uživatelé si rychle zvykli na pohodlné barevné prostředí a díky tomu se začaly rozšiřovat TFT displeje (kap. 3.1), čemuž se museli přizpůsobit i ostatní výrobci.

Na začátku rozvoje TFT displejů bylo ale připojení k mikrokontroléru (či procesoru) poměrně problematické. Bylo nutné používat specializované signálové procesory (DSP), hradlová pole (FPGA), či jednocelové řadiče. Pro větší rozlišení a kvalitnější grafiku bylo nutné využívat grafickou kartu, či mobilní grafické akcelerátory. Mezi nejznámější výrobce grafických karet patří firmy nVidia a ATI, nejrozšířenější mobilní akcelerátory zahrnují například řady Qualcomm MSM, Intel XScale, Texas Instruments OMAP, Imagination Technologies PowerVR a Nvidia Tegra.

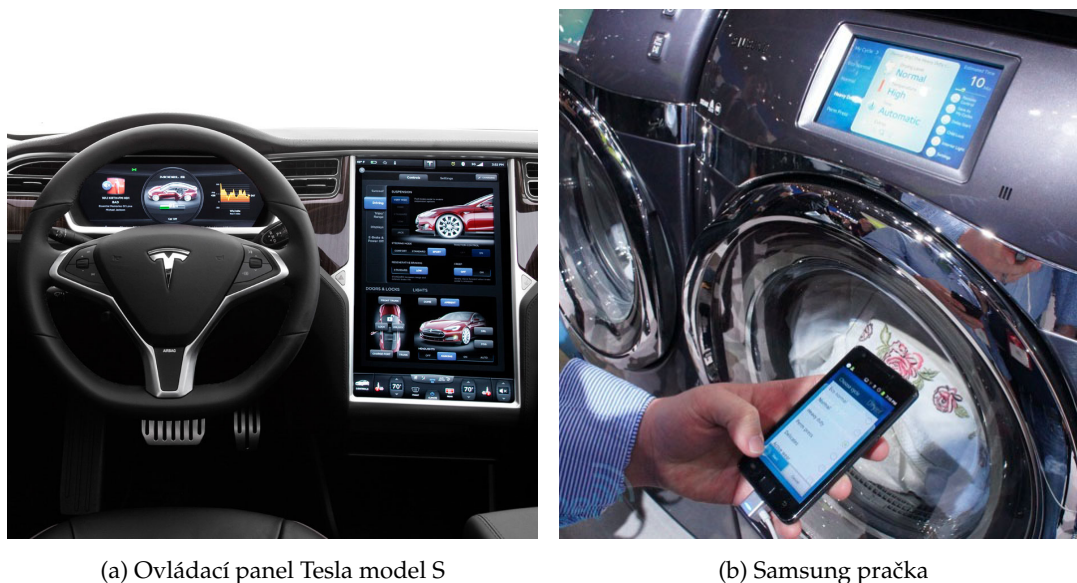
Mezi rozšířené řadiče pro vestavěné zařízení se řadí mimo jiné FPGA platforma „Altera Cyclon“ s procesorovou architekturou „NIOS“. Pro FPGA existují i editory uživatelského rozhraní, zdařilý se zdá být Altia Design [46]. Příkladem vývojové desky s čipem FPGA je modul Thomas II od firmy Altera [47].

## 1.1 SOUČASNÉ TRENDY A BUDOUCÍ VÝVOJ DISPLEJŮ

Dnešní moderní řešení používají především TFT-LCD, nebo AMOLED displeje (kap. 3.1) s úhlopříčkami okolo 5 – 7 palců. Nejčastější rozlišení vestavěných displejů bývají:

- VGA (640 × 480)
- WVGA (800 × 480)
- SVGA (800 × 600)
- XGA (1024 × 768)

Mezi současné trendy patří používání TFT LCD i v odvětvích, kde to dříve nebylo běžné. Příkladem může být například obří dotykový panel nainstalovaný v automobilech Tesla S (obr. 2a), či inteligentní pračka společnosti Samsung (obr. 2b). Mezi známé moderní koncepty patří ale například i termostaty Nest, které odkoupila společnost Google.



Obrázek 2: Současné trendy ve využití TFT displejů

Jemnost displejů je udávána v pixelech na palec čtvereční. Pohybuje se většinou v hodnotách mezi 100–300 PPI (pixels per inch). Pro představu, tablety iPad 1 a 2 generace dosahují rozlišení XGA při úhlopříčce 9,7 palce, čemuž odpovídá 132 PPI. Použitý displej v této práci [5] má úhlopříčku 7" při WVGA rozlišení, hustota je tedy 130 PPI.

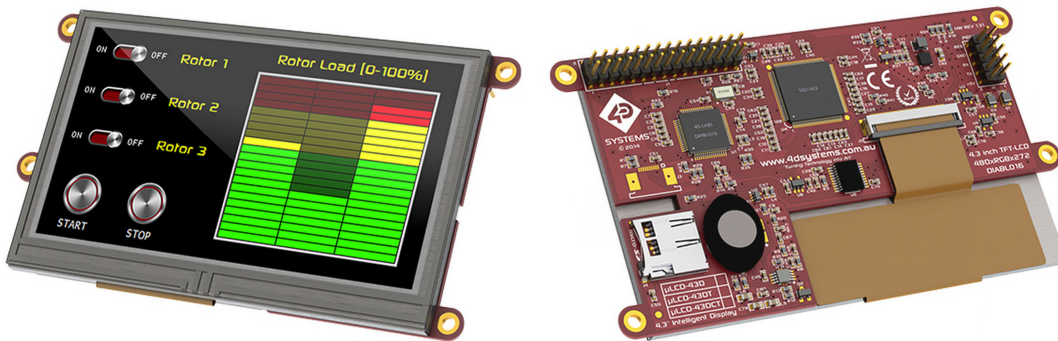
Od roku 2010 byl zaveden pojem „retina“. Z počátku se jednalo o marketingové označení společnosti Apple, definující minimální jemnost vzhledem ke vzdálenosti pozorovatele od displeje. Jednotkou je PPD („pixels per degree“) a bylo stanoveno, že označení retina bude používáno od 57 PPD. Pojem postupně přebrali všichni velcí výrobci zařízení na trhu.

Do budoucna se počítá s trendem zvyšováním jemnosti. V mobilech se dnes často používá rozlišení FHD (1920 × 1080) až WQHD (2560 × 1440) odpovídající jemnosti přes 400 PPI. Nejnovější generace tabletů dosahují rozlišení 4K UHD (3840 × 2560). Extrém oznámila společnost Sharp, která hodlá v roce 2016 uvést na trh 5,5" displej s rozlišením 4K UHD (806 PPI) [48].

## 2 DOSTUPNÁ ŘEŠENÍ PRO TFT DISPLEJE

Na trhu již samozřejmě existují řadiče pro displeje vestavěných zařízení, které se v praxi používají. Mezi nejznámější patří například firma 4D Systems [28], která dodává hotové moduly za poměrně rozumné ceny. Kompletní řešení se pohybuje v přepočtu okolo 3000 Kč – 5500 Kč bez DPH, v závislosti na velikosti displeje a typu dotykové vrstvy. Firma nabízí i speciální kulaté panely, určené především do automobilů, jako náhradu za mechanické ukazatele. Dodáván je i obslužný software. Dobře prodávaný typ je např. uLCD-43 na obr. (3).

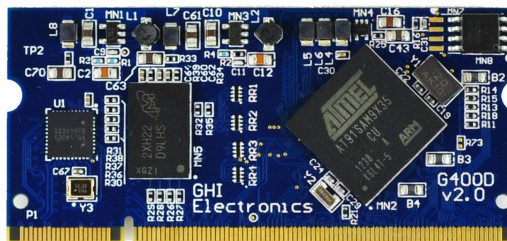
Nevýhodou je absence větších modelů v sortimentu, kdy většina displejů má úhlopříčku 0,96" až 4,3". Ve velikosti 7" je v nabídce jediný typ uLCD-70DT. Bohužel, všechny řadiče jsou pevně svázané s konkrétním typem panelu, který je vždy neoddělitelně připevněný. Je tak vyloučena možnost řadič použít na libovolný, vlastní typ panelu, což je jeden z přínosů této diplomové práce (viz kapitoly 5.1 a 5.3).



Obrázek 3: Řadič uLCD-43 od firmy 4D Systems

Vynikající moduly SoM (System on Module) dodává firma GHI Electronics [29]. Jedná se o řešení určené pro embedded aplikace, které je možné přímo připojit na vlastní desku s periferiemi. Výkon je velmi uspokojivý, jsou použity 400 MHz 32-bit mikrokontroléry ARM 9 řady SAM9X35 od firmy Atmel. Velkou výhodou u modulu G400-S na obr. (4) je dostatek RAM, kapacita je 92 MB. Cena se pohybuje okolo 2000 Kč bez DPH.

Ne všem zákazníkům ale může vyhovovat programování v .NET Micro Framework (NETMF). Omezením pro graficky náročnější aplikace může být i výstup pro displej, který umožňuje připojení maximálně RGB454 (13 bit), takže nejrozšířenější panely s RGB565 (16 bit), RGB666 (18 bit) a RGB888 (24 bit) nejsou maximálně využité.



Obrázek 4: Modul G400-S od firmy GHI



## 3 TEORETICKÁ ČÁST

### 3.1 TYPY ZOBRAZOVACÍCH TECHNOLOGIÍ

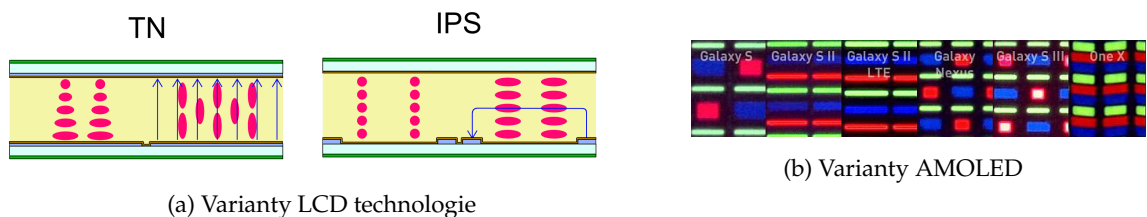
Technologie LCD (Liquid Crystal Display) stanovuje způsob, jakým jsou vyrobeny sub(pixely) zobrazovacího panelu. Jde o matici obsahující tekuté krystaly, které jsou prosvěcovány světlem skrz dva průhledné polarizační filtry. Tekutý krystal leží mezi filtry a ty mají vůči sobě otočenou rovinu polarizace o  $90^\circ$ . Bez krystalů by mezi filtry neprošlo žádné světlo. Elektrody v každém pixelu ale vytvářejí elektrické pole, které ohýbá molekuly krystalů do různých struktur, což stáčí i procházející světlo. Tím je možný průchod přes oba polarizační filtry (obr 5a). Informace byly čerpány ze zdroje [40].

Pojem TFT (Thin Film Transistor) označuje způsob, jakým je vyrobena budicí vrstva pod jednotlivými pixely. Jedná se o aktivní tenkou vrstvu tranzistorů, kdy každý ovládá jednotlivé subpixely. TFT vrstva je podobná všech LCD monitorů a kvalita zobrazení je určena právě technologií, jakou jsou vyrobeny jednotlivé (sub)pixely.

V současné době (2015) na trhu existují tři hlavní technologie a jejich deriváty, které se používají ve vestavěných systémech.

- TN (Twisted Nematic). Vyznačuje se rychlou odezvou a nízkými náklady, ale opticky není příliš kvalitní (bitová hloubka, pozorovací úhly). Odezva je v řádu jednotek milisekund. Výrobce použitého displeje ET0700G0DH6 [5] v této práci uvádí přechod z bílé do černé okolo 10 ms a pozorovací úhly až  $65^\circ$ . Věrné zachování barev je ale pouze při přímém pohledu pod úhlem do  $5^\circ$ . Většina TN panelů používá šestibitovou hloubku na kanál.
- IPS (In Plane Switching). Princip je podobný jako u TN, ale krystaly jsou umístěny vzhledem k polarizačním filtrům rovnoběžně, místo kolmému uspořádání. Technologie byla vyvinuta firmou Hitachi a později použita i u LG Display. IPS podporuje větší pozorovací úhly, větší kontrast a barvy jsou věrnější. Rozšířily se deriváty jako AS-IPS, IPS-Pro, H-IPS a e-IPS. Stále probíhá vývoj a zlepšuje se kontrast, jas a elektrická náročnost.
- AMOLED (Active Matrix Organic Light Emitting Diode). AMOLED panely vynikají vysokým rozlišením a dobrým podáním barev. Mají vysokou obnovovací frekvenci a ostrost. Skládají se z vrstvy organických LED diod, která je spínána TFT vrstvou, podobnou jako u technologie TN a IPS. Firma Samsung tuto technologii ještě vylepšila a pojmenovala Super AMOLED.

Použitý mikrokontrolér v této práci je vybaven periferií LTDC (LCD-TFT Display Controller, kap. 4.5). Tato periferie (řadič) umožňuje připojení a časování panelu (kap. 4.6), zároveň implementuje prolínání a kvantování barev. Právě díky tomu je možné připojit TFT LCD displej bez dalších externích součástí. Zdroj obrázků níže je [41].



Obrázek 5: Různé zobrazovací technologie

## 3.2 KÓDOVÁNÍ BAREV

Obrazová data jsou LCD řadičem na displeji reprezentována pomocí pixelů. Existuje mnoho barevných prostorů, přičemž každý definuje hodnotu jednoho pixelu jiným způsobem (intenzita kanálů, jas, průhlednost; reprezentace pomocí hodnot integer nebo float). Úrovně barev musí být při binární reprezentaci kvantovány, takže zobrazení výsledné barvy (polotónu) je pokaždé jinak přesné (spektrum není po kvantizaci spojitě). Kvantovací chyba se dá odstranit přidáním pseudonáhodným šumem (dither) pomocí předem daného algoritmu pro generování. Tato metoda se jmenuje dithering [18]. Přehled nejčastějších barevných prostorů viz [17].

Barevné prostory podporované řadičem v použitém mikrokontroléru [1] jsou zobrazeny v tabulce (tab. 1). Jednotlivé polotóny jsou nejčastěji reprezentovány přímo pomocí intenzit kanálů red (R), green (G) a blue (B). Některé barevné prostory umožňují definovat průhlednost A (alpha), což je velmi výhodné v situacích, kdy je třeba zobrazovat objekty přes sebe. Použitý řadič průhlednost i dithering podporuje nativně na hardwarové úrovni (kap. 4.5, obr. 17). S tím úzce souvisí možnost klíčování, kdy je určitý barevný polotón nahrazen jiným. Takto může být například odstraněno jednobarevné pozadí textu či obrázků a nahrazeno průhledným. Některé barevné prostory dokáží definovat i složku jasu L (luminance), což je využíváno například u jednobarevných pozadí.

Jednotlivé prostory aproximují obrazovou informaci do přirozených násobků bytů. Proto je reprezentace pomocí jednoho bytu (L8, AL44), dvou bytů (RGB565, ARGB4444, AL88, ARGB1555), tří bytů (RGB888), či čtyř bytů (ARGB8888). Pro zjednodušení byly zavedeny pojmy High color pro 16 bit barvy a True color pro 24 bit barvy. Displeje často implementují 18 bitové připojení (3 × 6 bit), takže je výhodné pro přesnější reprezentaci barev používat RGB888 či ARGB8888 prostor a spodní bity zahodit (kap. 3.3).

Parametr	R	G	B	Polotónů	A	L
ARGB8888	256	256	256	16 777 216	256	0
RGB888	256	256	256	16 777 216	0	0
RGB565	32	64	32	65 536	0	0
ARGB1555	32	32	32	32 768	2	0
ARGB4444	16	16	16	4 096	16	0
L8	0	0	0	0	0	256
AL44	0	0	0	0	16	16
AL88	0	0	0	0	256	256

Tabulka 1: Podporované barevné prostory

Důvod, proč některé barevné prostory obsahují dvojnásobek informací o zeleném kanálu (např. RGB565 – 32 červených a modrých polotónů, 64 zelených), vychází z vlastností lidského oka. Člověk je evolučně přizpůsobený na život v přírodě, kde se zelená vyskytuje ve velké míře a správná interpretace v minulosti znamenala vyšší pravděpodobnost přežití. Lidské oko je na zelenou nejvíce spektrálně citlivé, což reflektuje taktéž barevný gamut Adobe RGB oproti gamutu sRGB.

Bayerova maska ve snímačích fotoaparátu má právě proto dvojnásobek zelených polí. Dokáže ale zaznamenat pouze jasy jednotlivých buněk a výslednou barvu skládá pomocí interpolace sousedních buněk. Podobný princip je využíván v TFT displejích, kdy horizontální řádek obsahuje trojnásobek jednotek tvořících světlo (subpixelů), oproti efektivnímu horizontálnímu rozlišení. Odlišný přístup využívá ve fotografii čip Foveon od společnosti Sigma, kdy jsou pixely tvořeny na základě vlastností pronikání jiných vlnových délek světla do různých hloubek křemíku.



### 3.3 CHYBY KÓDOVÁNÍ

Chybný převod mezi formáty může způsobit problémy se zobrazením, případně kolaps celé obrazové informace. Barvy mají být správně kódovány tak, že informace začíná nejdůležitějším bitem (MSB) a při konverzi z vyšší bitové hloubky na nižší se zahazují nejméně důležité bity, tedy spodní (LSB). Tento proces se anglicky označuje jako „pixel format conversion“ (PFC).

Může však nastat situace, kdy vývojář nereflektuje standard a konverze se provede špatně. Problém může nastat ale i v návrhu hardware. Řadič displeje má  $3 \times 8$  bitů výstup pro kanály RGB. Velké množství displejů ale používá 6 bitů na kanál, případně hloubku RGB565. Jako vhodné řešení se na první pohled zdá propojit vodiče R[5..0] s vodiči R[5..0], analogicky pro G a B složku. Taková úvaha je ale chybná a je nutno brát v potaz, že se takto ztratí právě dva nejdůležitější bity barvy. Stejná situace nastala při konstrukci redukce (kapitola 5.2) po použití displej [5] v této práci. Řešením je vytvoření správné redukce, případně programovatelné mapování pinů na displeji, což je hlouběji rozpracováno v kapitole 5.3.

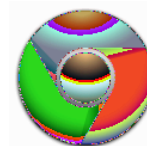
Kolaps obrazové informace je dále zobrazen na obrázcích obr. (6) a (7), což byl problém první verze redukce (kap. 5.2). Chybná interpretace byla simulována skriptem bmp2c [19]. Barvy setímto způsobem mohou změnit, kdy se zelená téměř zachovala, žlutá se změnila na červenou a červená na modrou. Modrá se rozdělila na bílou a černou (byla definována jenom horními bity). Obrazová informace je ale stále patrná. Tabulka (tab. 2) ilustruje výše jmenované chyby na vybraných barvách u ikony prohlížeče Google Chrome zahazením horních dvou bitů při převodu na 6 bitové číslo.

	RGB888		RGB666
zelená	0x4B, 0xB6, 0x47	→	0x0B, 0x36, 0x07
žlutá	0xFC, 0xD2, 0x09	→	0x3C, 0x12, 0x09
červená	0xE2, 0x3D, 0x35	→	0x22, 0x3D, 0x35

Tabulka 2: Chybná interpretace barev zahazením horních dvou bitů



(a) 8 bitů na kanál



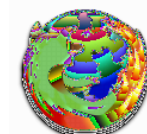
(b) dolních 6 bitů na kanál

Obrázek 6: Ikona prohlížeče Google Chrome

Horší varianta nastala u komplikovanější bitmapy (ikona prohlížeče Firefox) na obr. (7), kdy je původní informace jenom obtížně interpretovatelná.



(a) 8 bitů na kanál



(b) dolních 6 bitů na kanál

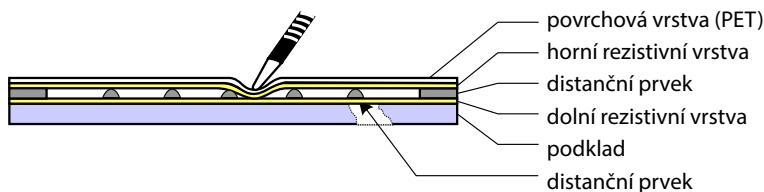
Obrázek 7: Ikona prohlížeče Mozilla Firefox

### 3.4 SNÍMÁNÍ DOTYKŮ

Pro lepší komfort ovládání byla implementována podpora dotykové vrstvy. Existuje několik technologií snímání a mezi nejvíce používané patří:

1. Kapacitní – měří se změna kapacity.
2. Infračervená – měří se přerušení horizontálních a vertikálních infračervených paprsků.
3. Optická – z jedné strany se prosvěcuje zobrazovací plocha infračerveným světlem a na druhé straně se signál snímá a trianguluje.
4. Ultrazvuková – SAW, do displeje jsou vysílány ultrazvukové vlny, které dotyk prstu přeruší.

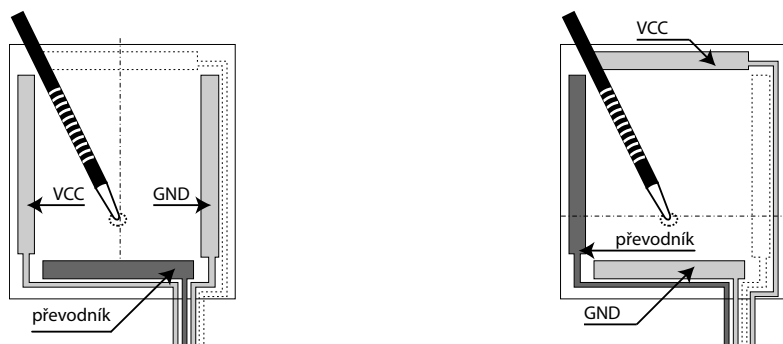
Velmi rozšířená je ale i rezistivní technologie, na jejímž principu funguje dotyková vrstva použitého displeje v této práci ET0700G0DH6 [5]. Nejčastěji se používá 4-, 5- a 8-vodičové připojení, přičemž použitá varianta je čtyřvodičová. Princip fungování je ilustrován následujícím obrázkem (obr. 8). Zdroj originálů obrázků je [36].



Obrázek 8: Princip fungování dotykové vrstvy

Dotyková plocha se skládá z dvou odporových vrstev pokrytých vrstvou ITO (oxid inditý a cínčitý), které se dotykem propojí. Proti náhodnému propojení je plocha vybavena drobnými distančními prvky, nejčastěji podobným kuličkám. Po stranách jsou čtyři vodivé plochy (v každé odporové vrstvě dva proti sobě), na které se postupně napojuje napájecí napětí, zem a měřicí elektroda. Čtyřvodičové připojení přivádí ke každé ploše jeden vodič, osmivodičové ke každé ploše dva (měření pak není ovlivněno protékajícím proudem). Pětivodičové připojení přidává jeden společný středový vodič „sense“ pro převodník.

Princip vyčítání polohy je patrný z obrázků (9a a 9b). Na protilehlé strany jedné rezistivní vrstvy se vždy připojí napájecí napětí, které je snímáno jednou z elektrod jiné vrstvy. Zbývající elektroda je vždy nevyužita. Tím je vyčtena jedna souřadnice. Snímání druhé souřadnice funguje analogicky. V kapitole 4.7 je dále popsán princip použitého specializovaného řadiče dotykové vrstvy.



(a) Snímání horizontální polohy

(b) Snímání vertikální polohy

Obrázek 9: Snímání polohy rezistivního dotykového displeje

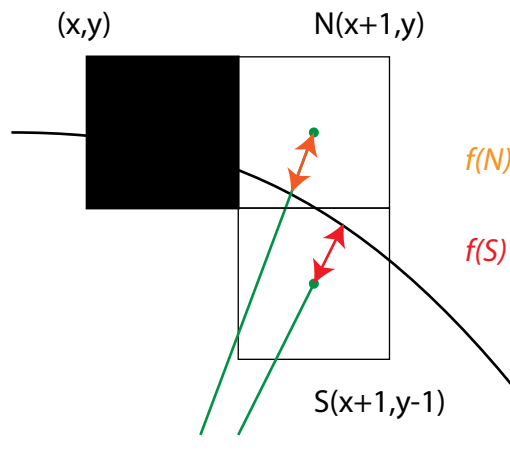
### 3.5 BRESENHAMŮV ALGORITMUS

Při vykreslování vektorových objektů na displej je třeba křivky správně aproximovat posloupností diskretních obrazových bodů (pixelů). Tomuto procesu se říká rasterizace vektoru. Je známých několik postupů, výhodné je použití například DDA (Digital Differential Analyzer), případně Bresenhamova algoritmu. Implementace je velmi efektivní a výpočetně málo náročná, protože inkrementální algoritmy fungují pouze na základě součtů a násobení čísel. Není nutné počítat goniometrické funkce, ani desetinná čísla. Pro aproximaci vektorů v této práci byl použit Bresenhamův algoritmus.

Algoritmus vymyslel Jack Elton Bresenham v létě 1962 pod hlavičkou firmy IBM. Je takto možné aproximovat přímky (úsečky), elipsy, Bézierovy křivky a kružnice. Varianta pro vykreslování kružnic je často nazývána také „midpoint circle algorithm“ a v kapitole (kap. 3.5.1) následuje odvození (čerpáno z [20]). V rámci této práce byl implementován základ dodávaný od firmy ST Microelectronics, dopsána byla varianta pro kreslení jednotlivých vybarvených i nevybarvených oktantů kružnic. Toho je využito například při kreslení obdélníků se zaoblenými rohy.

#### 3.5.1 ODVOZENÍ

Je uvažován kousek vektorového oblouku, který je potřeba vyjádřit rastrově rozsvěcováním pixelů (obr. 10). Předpokládá se, že byl právě vykreslen pixel  $(x, y)$  a nyní přichází v úvahu dva další pixely, z nichž právě jeden bude dále vykreslen. Ty jsou označeny jako „severní“ (N) a „jižní“ (S). Proto jsou zavedeny funkce  $f(N)$  a  $f(S)$ , které vrací odchylku oblouku od středu zvažovaného pixelu. Směrem ke středu oblouku je odchylka záporná, na opačnou stranu kladná.



Obrázek 10: Rasterizace vektoru pomocí Bresenhamova algoritmu

Zavedením rozhodovacího parametru  $d_i$  jakožto součtu odchylek

$$d_i = f(N) + f(S) \quad (1)$$

se dále rozhoduje o dalším vykresleném pixelu.

Pokud je parametr  $d_i \leq 0$ , znamená to, že je „severní“ (N) pixel blíže ke kružnici, tedy

$$x_{i+1} = x_i + 1, \quad y_{i+1} = y_i. \quad (2)$$

Naopak, bude-li parametr  $d_i > 0$ , „jižní“ pixel (S) je blíže, takže

$$x_{i+1} = x_i + 1, \quad y_{i+1} = y_i - 1. \quad (3)$$

Rovnice kružnice je

$$x^2 + y^2 = r^2, \quad (4)$$

kde  $r$  značí poloměr kružnice, tedy vstupní hodnotu algoritmu.

Odchylky středu „severního“ a „jižního“ pixelu od oblouku je tedy možné vyjádřit dosazením (2) a (3) do (4). Jsou tedy získány funkce  $f(N)$  (5) a  $f(S)$  (6).

$$f(N) = (x_i + 1)^2 + y_i^2 - r^2 \quad (5)$$

$$f(S) = (x_i + 1)^2 + (y_i - 1)^2 - r^2 \quad (6)$$

Dosazením vztahů (5) a (6) do (1) je vyjádřen rozhodovací parametr současného kroku pro kružnici jako

$$d_i = 2(x_i + 1)^2 + y_i^2 + (y_i - 1)^2 - 2r^2. \quad (7)$$

Rozhodovací parametr pro další krok  $d_{i+1}$  bude tedy

$$d_{i+1} = 2(x_i + 2)^2 + y_{i+1}^2 + (y_{i+1} - 1)^2 - 2r^2. \quad (8)$$

Rozdílem (8) a (7) je získána diference mezi jednotlivými kroky. Za povšimnutí stojí, že tento vztah už není závislý na poloměru kružnice  $r$ .

$$d_{i+1} - d_i = 2((x_i + 2)^2 - (x_i + 1)^2) + (y_{i+1}^2 - y_i^2) + ((y_{i+1})^2 + (y_i - 1)^2) \quad (9)$$

Vhodnou úpravou (9) je získán vztah (10), nyní teď už bez proměnné poloměru  $r$ .

$$d_{i+1} = d_i + 2(2x_i + 3) + (y_{i+1} + y_i)(y_{i+1} - y_i) + ((y_{i+1} - 1 + y_i - 1)(y_{i+1} - 1 - y_i - 1)) \quad (10)$$

Pro  $d_i \leq 0$ , tedy „severní“ pixel platí (2). Po dosazení do vztahu výše (10) a úpravě je

$$d_{i+1} = d_i + 4x_i + 6. \quad (11)$$

Analogicky pro „jižní“ pixel s ohledem na (3) po dosazení do (10) a úpravách je

$$d_{i+1} = d_i + 4(x_i - y_i) + 10. \quad (12)$$

Tím je odvození dokončeno; výsledný algoritmus je založen právě na vztazích (11) a (12), mezi kterými se v cyklu rozhoduje. Tak je aproximován oblouk, v tomto případě jeden oktant kružnice. Pro zbytek oktantů se využívá symetrie kružnice (podle os  $x$  a  $y$  a podle diagonál v  $\pm 45^\circ$ ), takže při vykreslování se pracuje vždy s osmi body najednou.

Je třeba znát ještě počáteční podmínku. Je zvykem začínat „nahore“ kružnice, tedy v bodě  $x = 0$  a  $y = r$ . Dosazením do (7) je tedy vyjádřena počáteční podmínka jako  $d_0 = 3 - 2r$ .

Při použití Bresenhamova algoritmu na vykreslení přímk je třeba si uvědomit, že je založen na rozdílu diferencí. Vertikální přímka má ale nekonečnou derivaci (diferenci) a na ní algoritmus selže. Funkce sloužící k vykreslení horizontální a vertikální linky jsou z tohoto důvodu implementovány zvlášť. Je využito možností DMA2D, kdy periferie „chápe“ oblast displeje a linku dokáže na základě hardwarové akcelerace rychle a efektivně dopočítat.

### 3.6 VNITŘNÍ STRUKTURA BMP OBRÁZKU

Častým požadavkem z pohledu uživatele je podpora zobrazení obrázků. Proto byl pro implementaci v příložených ovladačích Board Support Package (kap. 6.2) vybrán primárně formát BMP. Výhoda je jednoduchost – data jsou reprezentována v nekomprimované podobě přímo pomocí pixelů. Na začátku každého BMP souboru je hlavička o velikosti 54 bytů, která obsahuje informace o obrázku (tab. 3).

Bezprostředně za hlavičkou následují obrazová data, která mohou být kódována pomocí různé bitové hloubky (1 bit – 2 barvy na pixel, 4 bity – 16 barev, 8 bitů – 256 barev, 16 bitů – 65 535 barev, nebo 24 bitů – 16,7 milionu barev). True Type (24 bit) varianta byla implementována i pro tvorbu snímků obrazovky k příloženým kódům (kap. 6.3).

Offset	Velikost	Význam
0	2	Signatura, musí být 0x4D42 („BM“).
2	4	Velikost BMP obrázku v bytech (nespolehlivé).
6	2	Rezervováno, musí být 0.
8	2	Rezervováno, musí být 0.
10	4	Offset do začátku obrázku v bytech (54).
14	4	Velikost hlavičky, musí být 40.
18	4	Šířka obrázku v pixelech.
22	4	Výška obrázku v pixelech.
26	2	Počet vrstev obrázku, musí být 1.
28	2	Počet bitů na pixel (24), jinak 1, 4, nebo 8.
30	4	Typ komprese (0 = není) jinak, 1=RLE-8, 2=RLE-4.
34	4	Velikost obrázku včetně hlavičky.
38	4	Horizontální rozlišení v pixelech na metr (nespolehlivé).
42	4	Vertikální rozlišení v pixelech na metr (nespolehlivé).
46	4	Počet barev v obrázku, nebo 0.
50	4	Počet důležitých barev v obrázku, nebo 0.

Tabulka 3: Vnitřní struktura BMP obrázku

Formát BMP byl představen v roce 1988 v rámci systému OS/2. Firma Microsoft tento formát dále rozšířila a zahrнула v operačním systému Microsoft Windows 3.0. BMP umožňuje implementovat RLE kompresi, která kóduje vstupní data tak, že vytváří dvojice (délka posloupnosti, hodnota).

Jednoduchý příklad RLE:

$$\text{WWWWWWBWWWWBBB} \rightarrow 7W1B5W3B. \quad (13)$$

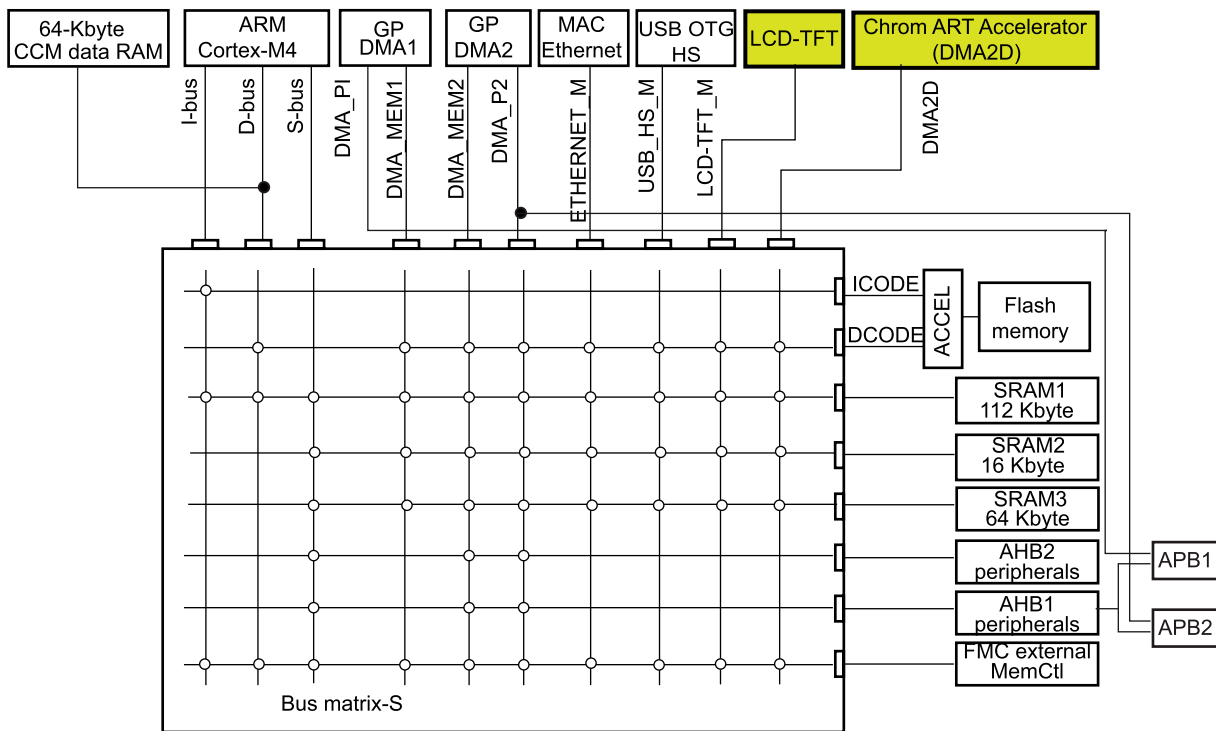
Zobrazení BMP obrázku je implementováno v BSP (Board Support Package) knihovně (kap. 6.2). Pro podporu formátů s komplikovanější kompresí (JPEG, PNG) je třeba použít některou z hotových profesionálních grafických knihoven pro STM32 procesory. To může být například STemWin (kap. 6.4.2), nebo Touch GFX (kap. 6.4.1).

### 3.7 ARCHITEKTURA MIKROKONTROLÉRU

Použitý mikrokontrolér STM32F439BIT6U patří do high-end MCU řady s jádrem Cortex M4. Podporované instrukce jsou Thumb, Thumb-2, DPS, SIMD a FP. Jádro Cortex M4 na rozdíl od Cortex M3 obsahuje hardwarovou jednotku pro práci s desetinnými čísly a podporu DSP (digital signal processing).

Vnitřně je čip postaven na rychlé sběrnici AHB (Advanced High-Performance Bus), která propojuje všechny další řadiče a periferie. Dělí se na zařízení typu „master“ (Cortex-M4 jádro, DMA1, DMA2, Ethernet, USB OTG HS) a „slave“. Mezi slave patří flash ICode a Dcode sběrnice, vnitřní SRAM1-3 a FMC (Flexible Memory Controller). Pomocí můstků jsou k maticové sběrnici AHB připojeny periferie na úspornějších a pomalejších sběrnících APB (USART, SPI, časovače, PWM, SDIO, ADC, DAC, RTC a další).

Řady STM32F4x7/9 mezi zařízení master přidávají řadič LTDC (LCD-TFT Display Controller) a speciální variantu převodníku DMA – Chrom-ART Accelerator (DMA2D). Právě na těchto periferiích je postavena grafická část řadiče. Architektura je zobrazena na obr. (11).



Obrázek 11: Maticová sběrnice 32-bit multi-AHB

Řadič postavený v této práci je optimalizován tak, aby dosahoval co největšího grafického výkonu. Obsahuje externí paměť SDRAM (kap. 4.9.4), která je připojena přes FMC. Do SDRAM přistupuje DMA2D, který dokáže paměť plnit velkou rychlostí daty určenými k vykreslení. Nabízí taky možnost konvertovat barevné formáty přímo při práci s pamětí (kap. 4.4, obr. 16).

O samotné zobrazení na displej se stará řadič LTDC (kap. 4.5). Umožňuje mezi sebou prolínat vrstvy (frame buffer) a grafická primitiva připravené díky DMA2D v paměti a stará se i o dithering a klíčování. LTDC obsahuje jednotku pro generování synchronního časování pro displej (kap. 4.6).

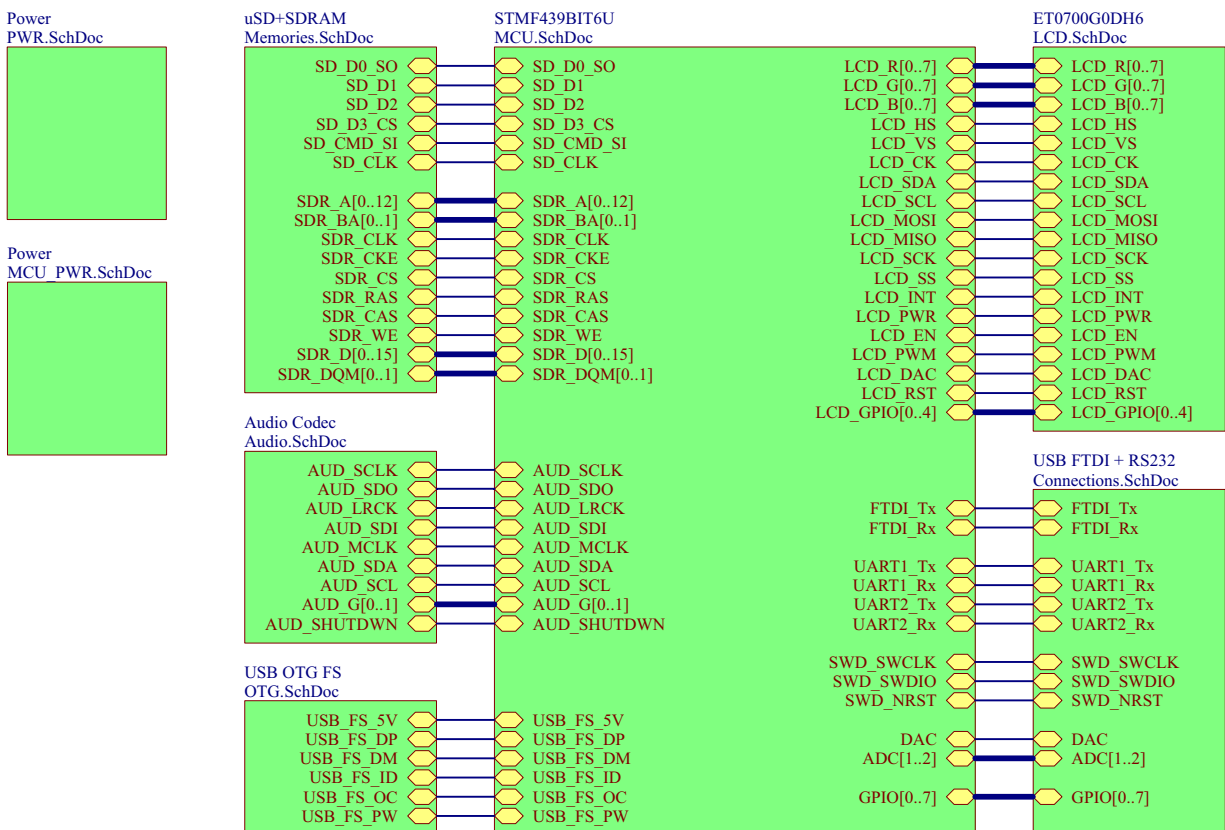
## 4 HARDWARE TFT LCD ŘADIČE

Pro návrh řadiče byl použit program Altium Designer. Software umožňuje vytvářet listy schémat (Sheets), z nichž jeden slouží jako hlavní a obsahuje rozvržení menších logických celků. Přípona souborů schémat je obvykle *SchDoc* a propojují se vodiči, nebo sběrnicemi. Z celého projektu je vygenerován soubor s plošným spojem a příponou *PcbDoc*.

Na obrázku obr. (12) je celkové schéma definované v hlavním listu *display.PcbDoc*. Nejdůležitější část je list *MCU.SchDoc* s mikrokontrolérem STM32F439BIT6U [4]. Mikrokontrolér je blíže popsán v kapitole 4.2, časování je vysvětleno v kap. 4.3. Definice filtračních kondenzátorů a napájení jádra je odděleno v souboru *MCU\_PWR.SchDoc*.

Externí paměti zahrnuje *Memories.SchDoc*, kde je patrné zapojení SDRAM (kap. 4.9.4) a microSD karty (kap. 4.8). Komunikace je možná pomocí FTDI (kap. 4.10.2) a RS-232 (kap. 4.10.1) a jejich schéma zahrnuje dokument *Connections.SchDoc*. USB OTG připojení (kap. 4.10.3) je definováno zvlášť v *OTG.SchDoc*.

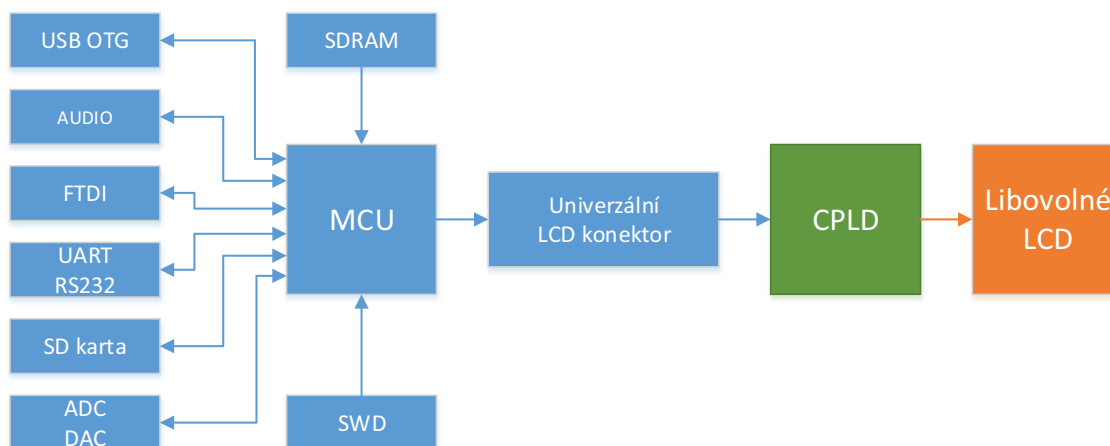
Audio část zahrnuje schéma *Audio.SchDoc*, využít je audiokodek WM8731 [11] (kap. 4.11.1) a zesilovač TPA2012 (kap. 4.11.4). Dokumenty jsou přiloženy jako součást této práce (viz příloha 10).



Obrázek 12: Organizace schémat

## 4.1 BLOKOVÉ SCHÉMA

Blokové schéma ilustruje následující obrázek (obr. 13).



Obrázek 13: Blokové schéma

Základem řadiče je mikrokontrolér STM32F439BIT6U [4] od firmy ST Microelectronics, který byl zvolen především pro dostatek GPIO pinů (pouzdro LQFP208) a přítomnosti LTDC periferie (LCD-TFT Display Controller), díky níž je možné nativně připojit TFT displej. Vykreslování grafických primitiv je akcelerováno pomocí DMA2D (Chrom-ART Accelerator). K programování je určen SWD konektor.

Mikrokontrolér je vyroben 90-nanometrovou technologií a je vybaven 2 MB Flash a 256 kB RAM. Paměť je možno rozšířit připojením na FMC řadič (Flexible Memory Controller). V rámci této práce byla použita 64 Mb SDRAM MT48LC4M16A2 [6] od firmy Micron, navržená deska ale podporuje všechny 16-bit varianty v pouzdře TSSOP2 54 pin až do velikosti 512 Mb. Paměť je využita především jako „frame buffer“ pro vykreslování grafických primitiv.

Pro uchovávání větších bitmap byla implementována podpora microSD karet. Vývojář firmware má možnost zvolit si komunikaci po rychlé SDIO 1/4 bit sběrnici, nebo pomocí standardního SPI. Souborový systém je podporován použitím knihovny FatFs [27].

Konektivita je vyřešena implementací USB OTG (On-The-Go) Fullspeed, možné jsou režimy Host i Device. Použit byl standardní konektor USB Micro AB, takže je možné připojit např. standardní flash disk. Pro posílání stavových zpráv a komunikaci s počítačem je osazen čip FTDI. Sériová komunikace je možná pomocí dvou portů UART/RS-232.

Deska je vybavena několika GPIO piny pro připojení dalších periférií, tlačítek, nebo LED. Další GPIO piny jsou přítomné na navrženém univerzálním LCD konektoru. Analogová část obsahuje 2× A/D a 2× D/A převodníky, přičemž jeden z D/A je přímo určen pro řízení jasu podsvícení panelu.

Interakce s uživatelem může probíhat taktéž zvukově, implementován byl audiokodek a dvoukanálový zesilovač. Díky tomu je možné přehrávat různé zvuky, nebo melodie při obsluze displeje.

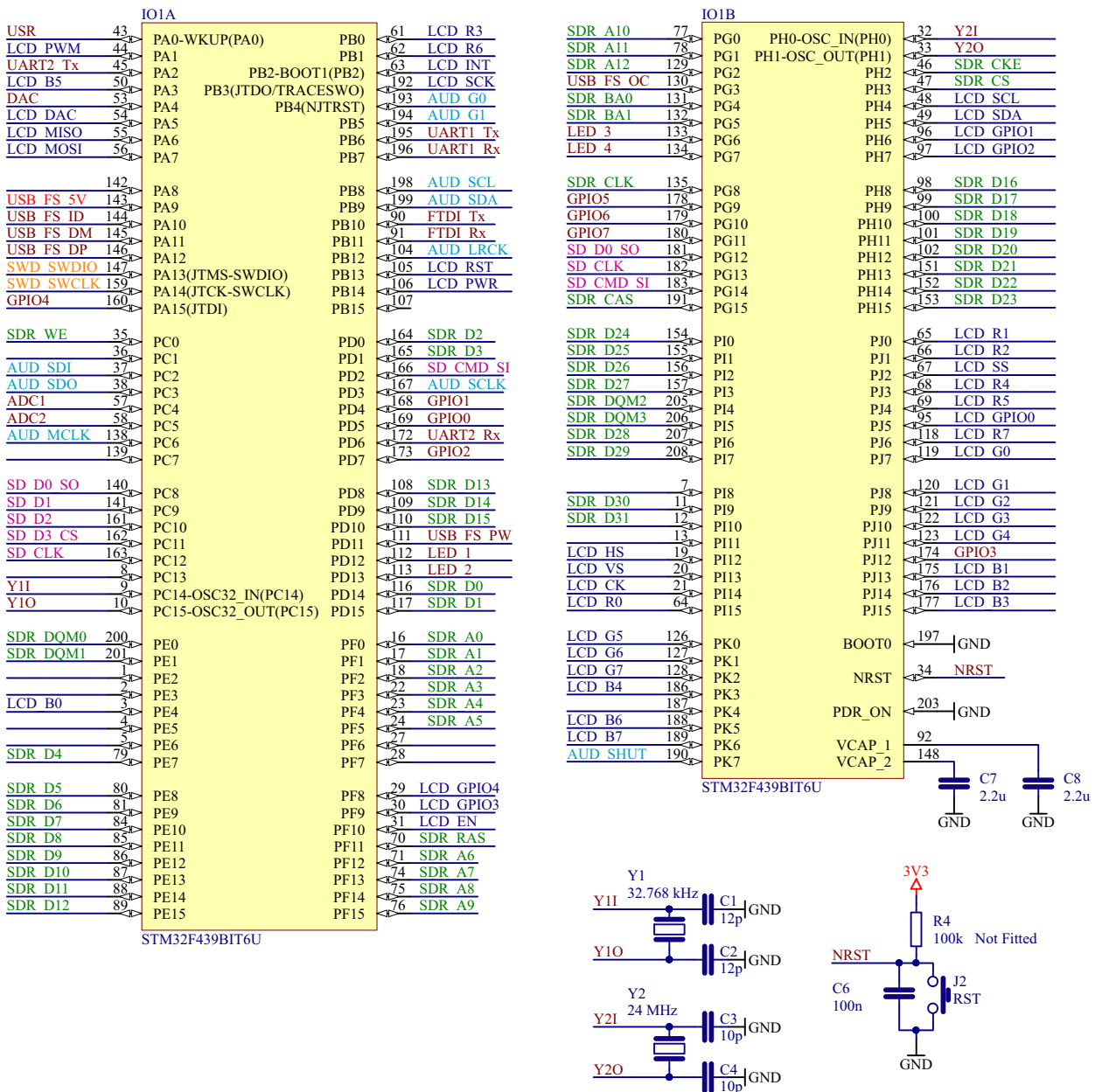
Přínosem této práce by měl být návrh univerzálního LCD konektoru (kap. 5.1). Díky němu je možné připojit libovolný displej, jehož podpora je realizována vždy pomocí pevné redukce (kap. 5.2), či díky programovatelné desce s CPLD (kap. 5.3).



## 4.2 MIKROKONTROLÉR

Pro účely diplomové práce bylo nutné s ohledem na zadání vybírat mikrokontrolér s podporou TFT LCD periferie. Toto kritérium splňují například řady procesorů STM32F429 a STM32F439 od firmy ST Microelectronics. Vzhledem k nutnosti připojení SDRAM, TFT LCD, USB, SD karty, GPIO a ostatních periférií byl zvolen model s pouzdrem LQFP208 se 169 GPIO piny.

Přesný typ použitého mikrokontroléru je STM32F439BIT6U z předprodukční série. Obsahuje Cortex M4 jádro a je vybaven 2MB flash pamětí a 256 kB RAM. Výkon je až 225 DMIPS při frekvenci 180 MHz. Mikrokontrolér je vyroben 90 nanometrovou technologií.



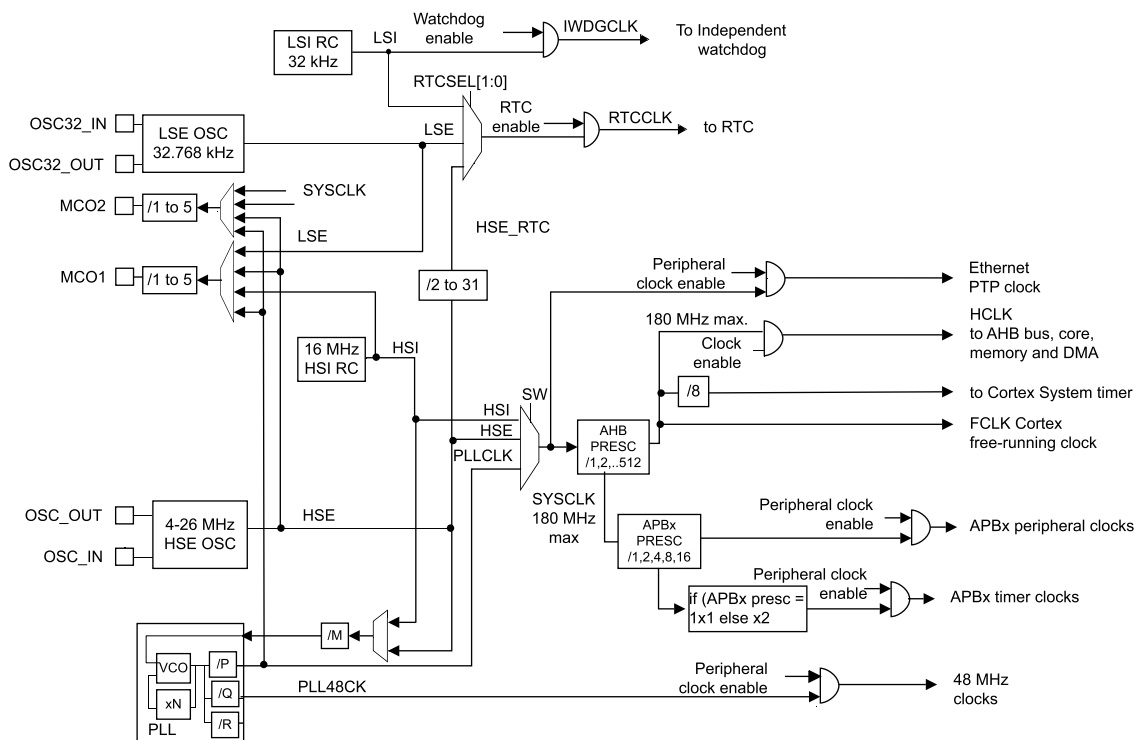
Obrázek 14: Mikrokontrolér STM32F439BIT6U

### 4.3 ČASOVÁNÍ MIKROKONTROLÉRU

Mikrokontrolér má vlastní interní krystal *HSI* („high-speed internal“) s frekvencí 16 MHz. Ten ale není velmi přesný, proto byl použit externí krystal *HSE* 24 MHz jako hlavní reference. Od něj se odvíjí veškeré časování. Pomocí PLL je třeba nastavit přesnou frekvenci 48 MHz pro SDIO sběrnici (kap. 4.8) a 16,125 MHz na LTDC periferii pro použitý displej ET0700G0DH6 [5] (kap. 4.5).

Bohužel, není možné nastavit zároveň maximální frekvenci 180 MHz a přesných 48 MHz pro SDIO sběrnici, žádná kombinace děliček a násobiček v PLL (Phase-Locked Loop) to neumožňuje. Proto musela být hlavní frekvence jádra a AHB sběrnice upravena na 168 MHz.

Obrázek (obr. 15) obsahuje strom časování. Hodinový signál 24 MHz z externího krystalu *HSE* je přiváděn do multiplexoru „PLL Source Mux“ a děličkou *PLLM* vydělen šestnácti, frekvence vstupující do PLL je tedy 1,5 MHz. Fázový záměr 1,5 MHz násobí hodnotou 224 (*PLLN*) na 336 MHz. Odsud je získán hodinový signál pro USB (*PLL48CK*) děličkou 7 (*PLLQ*) a pro hlavní jádro se nejvyšší frekvence dělí 2 na *PLLp*. Takto je tedy stanovena frekvence *SYSCLK* na 168 MHz.



Obrázek 15: Časování mikrokontroléru

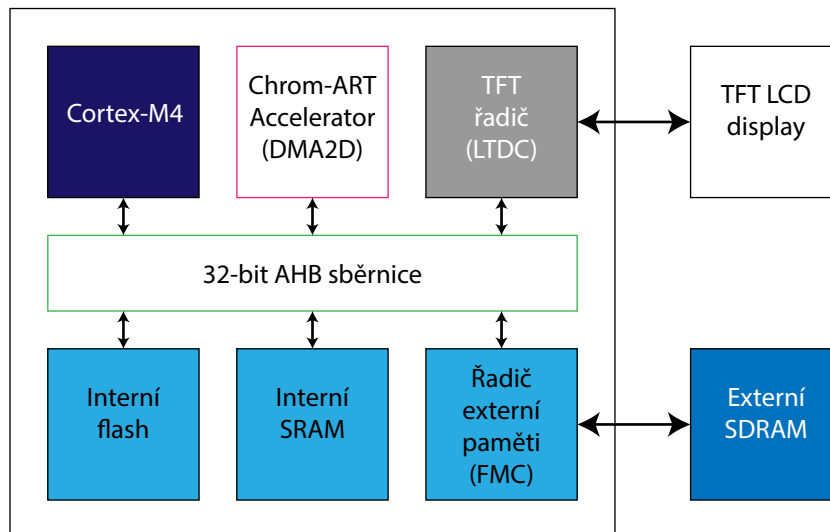
Další časování závisí na konkrétní aplikaci. Hlavní frekvenci *SYSCLK* je možné pro úsporu energie snížit předděličkou „AHB Prescaler“. V příložených demo kódech (kap. 6.3) toho ale není využito. K pomalejším sběrnicím *APB1* a *APB2* je taktéž přivedena maximální frekvence 42 MHz, resp. 84 MHz děličkami *APB1/2CLKDivider*.

Další informace ohledně oscilátorů a časování je možné nalézt v Aplikační poznámce výrobce [23].

## 4.4 GRAFICKÉ JÁDRO

Pro vykreslování grafiky byl dle zadání použit mikrokontrolér z řady STM32F429/F439, konkrétně byl zvolen typ STM32F439BIT6U [4]. Jedná se momentálně o nejvýkonnější variantu kontroléru z celé řady STM32F4xx. Právě díky periferiím LTDC a DMA2D (kap. 3.7 obr. 11) je práce s grafikou velmi efektivní.

Na obrázku (obr. 16) je výběr periferií pro vykreslování grafických primitiv. Nejdůležitější část je CPU, tedy procesorové jádro Cortex-M4. To slouží k výpočtu vektorů a uživatelského kódu. S ním úzce spolupracuje DMA2D (u ST Microelectronics označované jako Chrom-ART Accelerator), který přebírá práci s velkými objemy dat (např. obrázky), umožňuje konverzi barev, výpočet průhlednosti, klíčování a vybarvování objektů. DMA2D dokáže úzce spolupracovat s vnitřními paměti mikrokontroléru, ale i s řadičem externí paměti (FMC) a dokáže tak efektivně využívat i implementovanou externí SDRAM. Výsledný frame buffer je předán TFT řadiči (periferie LTDC), který se stará o zobrazení na displej. Periferie DMA2D je pro některé situace řádově rychlejší než procesorové jádro (přesun a konverze velkého množství dat, vyplňování objektů), proto je její použití velmi výhodné.



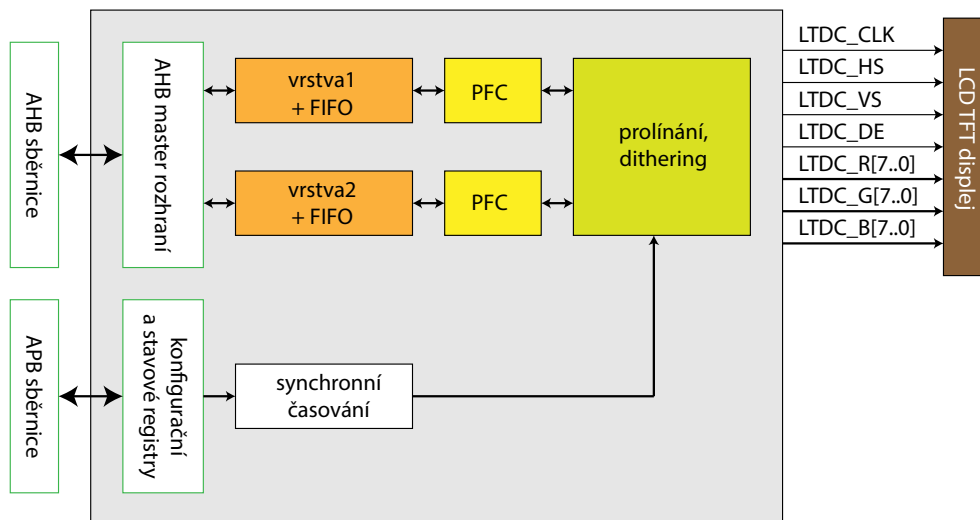
Obrázek 16: Architektura grafické části řady STM32F4x9

Standardní postup při práci s grafikou začíná inicializací použitých pinů TFT-LCD řadičem (LTDC) mikrokontroléru (kap. 5.1 tab. 26). Dále je nutné nastavit generátor synchronního časování periferie (kap. 4.5 obr. 18). Při inicializaci displeje je nutné nakonfigurovat rozlišení (kap. 4.6 tab. 5), polaritu signálu a barvu pozadí. Poté se alokují buffery pro obrazová data (frame buffery) v paměti SDRAM. Každý frame buffer může mít jinou velikost okna a jednotlivé vrstvy je možné prolínat, nebo klíčovat.

Vykreslení grafických primitiv probíhá zpravidla tak, že jsou pomocí DMA2D načteny velké bitmapy z externí paměti (USB OTG kap. 4.10.3 či microSD kap. 4.8) do externí SDRAM. Pro menší objemy dat (např. fonty) je vhodnější použít Cortex-M4 jádro, které je přesune z interní flash do SRAM. Všechna primitiva se pak sestaví a uspořádají do frame bufferů v externí SDRAM. Zobrazení na displeji zajistí LTDC, který jednotlivé frame buffery prolne, vyklíčuje, provede PFC (pixel format conversion) a dithering (odstranění kvantovací chyby vzniklé při PFC přidáním pseudoháhodného šumu). Libovolná změna ve frame bufferu SDRAM je okamžitě automaticky promítnutá na displej. Pro vymazání displeje nebo vybarvování velkých ploch je výhodné využít příslušnou vlastnost DMA2D.

## 4.5 LTDC PERIFERIE

Periferie LTDC (LCD-TFT Display Controller) řídí vykreslování obrazové informace na displej a stará se o časování, prolínání, kvantování barev a dithering. Obrazová informace se přenáší přes rychlou sběrnici AHB z paměti, přístup do registrů je realizován pomalejší sběrnici APB. Součástí periferie je generátor synchronního časování, dále popsany v kapitole 4.6.



Obrázek 17: Architektura periferie LTDC

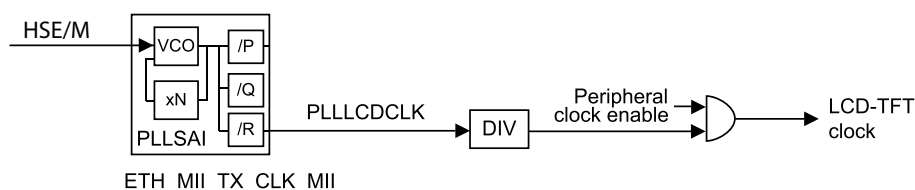
Periferie přímo podporuje tvorbu vrstev, které je možné nezávisle na sobě konfigurovat. Každá vrstva v paměti přistupuje k jednomu frame bufferu. První frame buffer je v příložených kódech alokovan na adrese 0xC000 0000, tj. počátek prostoru banky 5 pro FMC (Flexible Memory Controller), kde je připojena externí 64 Mbit SDRAM. Druhý frame buffer je obvykle na adrese

$$0xC000\ 0000 + \text{horizontální rozlišení} \cdot \text{vertikální rozlišení} \cdot \text{bitová hloubka.} \quad (14)$$

Při rozlišení  $800 \times 600$  použitého displeje ET0700G0DH6 [5] a barevném prostoru ARGB8888 (kap. 3.2) v paměti ( $3 \times 8$  bitů na pixel + 8 bitů průhlednosti) je velikost jednoho frame bufferu 1,536 MB.

Hodinový signál pro časování je získáván z externího krystalu *HSE* 24 MHz pomocí předděličky *M*. Je násoben vlastní PLL násobičkou hodnotou  $PLLSAIN = 258$  a dělen hodnotou  $PLLSAIR = 6$  (vztah 15). Tím je dosažena přesná frekvence 16,125 MHz pro vykreslování 30 snímků/s (vztah 19).

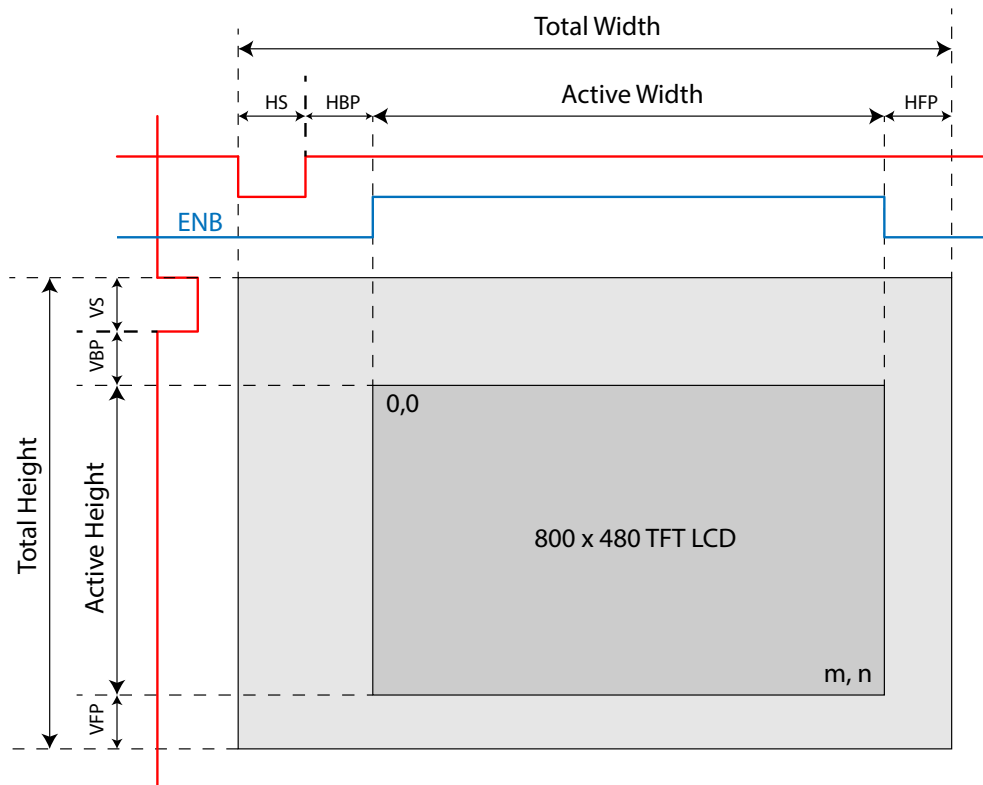
$$F_{CPH} = \frac{F_{HSE}}{PLLM} \cdot \frac{PLLSAIN}{PLLSAIR} = 33,25 \quad (\text{MHz}) \quad (15)$$



Obrázek 18: Časování LTDC periferie

## 4.6 ČASOVÁNÍ VIDEOSIGNÁLU

Časování displeje se odvíjí od standardního časování VGA (Video Graphics Array). Tento standard vyvinula v roce 1987 firma IBM. Od té doby je přímo či nepřímo podporován drtivou většinou zařízení, která dokáží pracovat s obrazem. Data jsou v případě VGA přenášena analogově vodiči Red (R), Green (G) a Blue (B) a obraz synchronizován vodiči *HSYNC* (horizontální synchronizace) a *VSYNC* (vertikální synchronizace). V případě TFT LCD displeje jsou barvy přenášeny digitálně, paralelně po sběrnicích R, G a B; správné vzorkování je zajištěno signálem *DCLK*. Použitý displej [5] obsahuje i vodič *ENB*. Podporovány jsou dva režimy řízení – vývojář vybere *HSYNC* a *VSYNC* variantu, nebo *ENB*.



Obrázek 19: Časování displeje ET0700G0DH6

Průběh signálu je i v případě LCD odvozen opět z historického standardu, kdy byla data vykreslována na CRT obrazovkách (katodových trubiciích) pomocí elektronového paprsku. Paprsek byl elektrostaticky nebo elektromagneticky vychylován cívkami a dopadal na fosforovou masku, se kterou interagoval a vytvářel obraz.

Průběh časování je rozdělen na horizontální a vertikální část. Signál horizontální synchronizace začíná impulsem *HSYNC* (HS) a následuje doba „Horizontal Back Porch“ (HBP), kdy paprsek doběhne na začátek následujícího řádku. Následují platná data, synchronizovaná signálem *DCLK* a potvrzovaná signálem „Data Enable“ (DE). Řádek končí úsekem „Horizontal Front Porch“ (HFP), kdy se čeká na další synchronizační impuls. Průběh vertikální synchronizace je obdobný.

Polarita synchronizačních pulsů je negativní, což znamená, že klidová hodnota je log. 1 a signál je aktivní při log. 0. Jinou kombinací polarit je v některých případech možné nastavit speciální režimy.

Základní DCLK frekvence ( $F_{CPH}$ ) použitého displeje je 33,26 MHz, takže perioda hodinového signálu ( $T_{CPH}$ ) je tedy 30,066 ns (16).

$$T_{CPH} = \frac{1}{F_{CPH}} = 30,066 \quad (\text{ns}) \quad (16)$$

Perioda jednoho řádku ( $T_H$ ) je součet zatmívacích částí na řádku, synchronizačního pulsu a aktivní doby řádku (17).

$$T_H = \text{HSYNC} + \text{HBP} + \text{WIDTH} + \text{HFP} = 1056 \quad (T_{CPH}) \quad (17)$$

Periodu potřebnou pro vykreslení jedné obrazovky ( $T_{FR}$ ) je možné vyjádřit vynásobením periody řádků se součtem vertikálních zatmívacích částí, vertikální aktivní doby a vertikálního synchronizačního pulsu (18).

$$T_{FR} = \text{VSYNC} + \text{VBP} + \text{HEIGHT} + \text{VFP} = 525 \quad (T_H) \quad (18)$$

Obnovovací frekvence ( $F_{REF}$ ) je vyjádřena podílem hlavní frekvence ( $F_{CPH}$ ) a periody jedné obrazovky ( $T_{FR}$ ). S ohledem na hodnoty tabulky (tab. 4) je to 30 Hz (19).

$$F_{REF} = \frac{F_{CPH}}{T_{FR}} \approx 30 \quad (\text{Hz}) \quad (19)$$

Parametr	Symbol	Hodnota	Jednotka
Horizontal Synchronization	HSYNC	128	$T_{CPH}$
Horizontal Back Porch	HBP	88	$T_{CPH}$
Active Width	WIDTH	800	$T_{CPH}$
Horizontal Front Porch	HFP	40	$T_{CPH}$
Vertical Synchronization	VSYNC	2	$T_H$
Vertical Back Porch	VBP	33	$T_H$
Active Height	HEIGHT	480	$T_H$
Vertical Front Porch	VFP	10	$T_H$

Tabulka 4: Časování pro displeje 800 x 480

Referenční signál pro periférii LTDC je získáván odvozením od frekvence externího krystalu *HSE* příslušnou PLL násobičkou (obr. 18). Do registrů periférie se nenastavují přímo hodnoty z tabulky (tab. 4), ale je nutné vypočítat potřebné kombinace těchto hodnot (tab. 5). Všechny hodnoty je ještě nutné dekrementovat o 1, jak je v tabulce naznačeno.

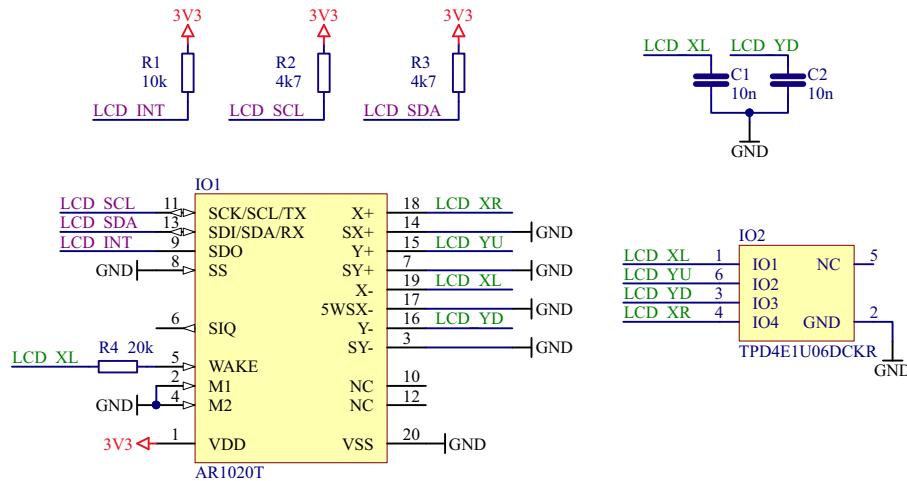
Parametr	Hodnota
Accumulated HBP	HSYNC + HBP - 1
Accumulated VBP	VSYNC + VBP - 1
Accumulated Active Height	HEIGHT + VSYNC + VBP - 1
Accumulated Active Width	WIDTH + HSYNC + HBP - 1
Total Height	HEIGHT + VSYNC + VBP + VFP - 1
Total Width	WIDTH + HSYNC + HBP + HFP - 1

Tabulka 5: Konfigurace LTDC periférie

Hodnoty pro použitý displej ET0700G0DH6 [5] jsou ve zdrojových kódech uloženy v hlavičkovém souboru ET0700G0DH6.h jako makra pro preprocesor. Použití jiného displeje je tedy možné realizovat pouhou změnou hlavičkového souboru.

## 4.7 ŘADIČ DOTYKOVÉ VRSTVY

Pro snímání dotyků na displeji byl využit specializovaný řadič AR1020T [13] pro rezistivní dotykovou plochu. Umožňuje 4-, 5- i 8-vodičové připojení, vzhledem k použitému panelu byla vybrána 4-vodičová varianta uzemněním nepoužitých signálů. Elektrické schéma je na obr. (20).



Obrázek 20: Schéma řadiče dotykové vrstvy

Dle doporučení výrobce byl implementován supresor napětí, který chrání řadič před výboji statické elektřiny, které by mohly nastat při kontaktu uživatele s dotykovou vrstvou. Byl zvolen obvod TPD4E1U06 firmy Texas Instruments. Jeden z vodičů je přiveden na pin WAKE, takže dotykem je možné probudit řadič z úsporného režimu, pokud jej bude nutno uspat.

Piny připojené k dotykové vrstvě jsou označeny *LCD\_XL*, *LCD\_XR*, *LCD\_YU* a *LCD\_YD* podle os (X, Y) a anglického pojmenování směrů („left“, „right“, „up“, „down“). V literatuře se ale může vyskytovat značení X+, X-, X+ a Y- (kladný směr je doprava a nahoru, záporný doleva a dolů). Řadič z obr. (20) je implementován zvlášť na redukci popsané v kapitole 5.2 a desce s CPLD (kap. 5.3) a nenachází se na hlavní desce, protože ne každý panel dotykovou vrstvou obsahuje.

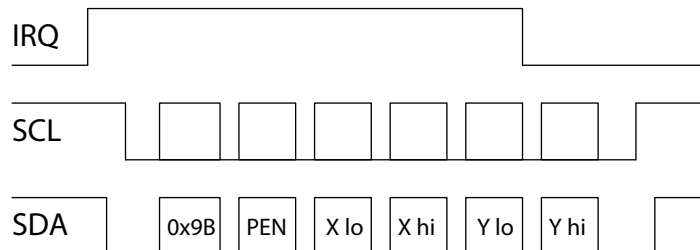
Řadič dotykové vrstvy AR1020T [13] měří napětí na čtyřech pinech pomocí 10-bitových A/D převodníků. Komunikace s mikrokontrolérem probíhá po sběrnici I2C. Dotyk je signalizován přepnutím pinu *LCD\_INT* z úrovně L do H, čímž je v mikrokontroléru vyvoláno přerušování a může následovat vyčtení souřadnic dotyku. Komunikační protokol a princip čtení je popsán v kapitole 4.7.1. Použité piny jsou vypsány do tabulky tab. (6).

Pin	MCU	Název
PH4	I1C2_SCL	LCD_SCL
PH5	I2C2_SDA	LCD_SDA
PB2	GPIO_Input	LCD_INT

Tabulka 6: Mapování vodičů pro touchscreen

#### 4.7.1 KOMUNIKAČNÍ PROTOKOL S ŘADIČEM

Podporu vyčítání souřadnic a stavu dotyku po I2C poskytuje „Touch report“ protokol. Zařízení komunikuje na adrese 0x9B a s příchozím přerušením na pinu *LCD\_INT* (signál *IRQ*) je nutné vyčíst 5 následujících bytů. Časový průběh je zobrazen na obr. (21).



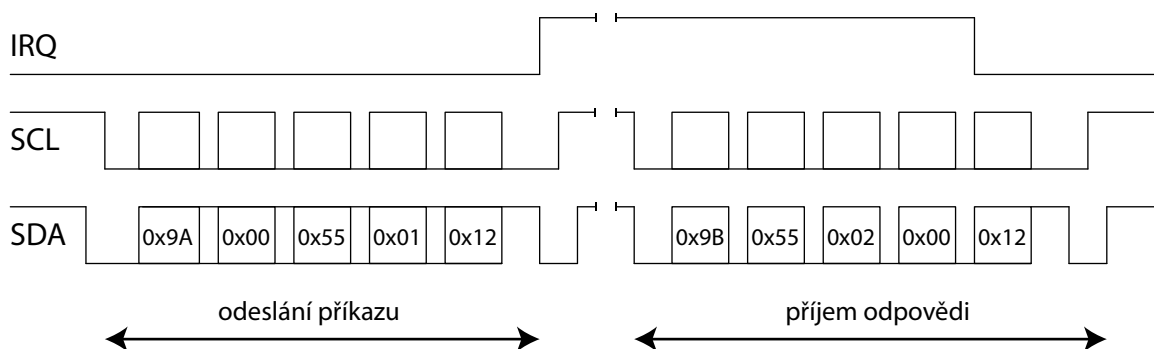
Obrázek 21: Touch report protokol

První byt paketu je označen „PEN“ a začíná bitem „1“ a končí bitem „P“, který signalizuje stav dotykové vrstvy (pera), kdy 0 znamená nahoře (nestisknuto) a 1 dole (stisknuto). Bity „R“ jsou rezervovány. Napětí z ploch na rezistivních vrstvách jsou interně měřena 10-bitovým A/D převodníkem, ale jsou zarovnána jako 12-bitová nahoru, přičemž dolní bity ( $X[1..0]$  a  $Y[1..0]$ ) jsou nulové. Tím je zajištěna budoucí kompatibilita s 12-bitovými převodníky. Podrobnější popis paketu „Touch report“ protokolu je v tabulce tab. (7).

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	R	R	R	R	R	R	P
2	0	X6	X5	X4	X3	X2	X1	X0
3	0	0	0	X11	X10	X9	X8	X7
4	0	Y6	Y5	Y4	Y3	Y2	Y1	Y0
5	0	0	0	Y11	Y10	Y9	Y8	Y7

Tabulka 7: Touch report protokol

Pro konfiguraci řadiče dotykové vrstvy je použit „Command protokol“, jejímž prostřednictvím je realizována mimo jiné i kalibrace (kap. 6.3.5). Příkaz se odesílá po I2C na adresu 0x9A a následuje „Protokol command byt“ 0x00, hlavička 0x55, velikost dat (0x01) a číslo příkazu, na obrázku obr. (22) je to 0x12 – „Enable touch“ příkaz. Čtení dat je realizováno po signalizaci přerušení z adresy 0x9B, kdy následuje hlavička 0x55, velikost 0x02, status 0x00 a číslo příkazu, na který se odpovídá.



Obrázek 22: Command protokol



## 4.8 PAMĚŤOVÁ KARTA MICROSD

Pro ukládání grafiky a uživatelských dat řadiče byla implementována podpora microSD karet. Secure Digital [25] (SD) obsahuje nevolatilní flash paměť – znamená to, že uchovává data i po odpojení napájecího napětí.

V zásadě se dnes rozlišují 3 typy karet – SDSC, SDHC a SDXC. Nejstarší typ je SDSC (Secure Digital Standard Capacity), který se dál dělá na verzi 1.xx a 2.xx. Tyto dvě verze se liší jenom počáteční inicializací. SDSC standard se dnes používá velmi okrajově, protože podporované jsou pouze kapacity 1 MB až 2 GB.

Nejvíce rozšířený standard je SDHC (Secure Digital High Capacity). Kapacita takovéto karty se pohybuje v rozmezí 4 GB až 64 GB. Není zpětně kompatibilní s velmi starými zařízeními a čtečkami, protože se oproti SDSC musí jinak inicializovat a navíc se používá jiné adresování dat. Zatímco SDSC karty používají bytovou adresu pro zápis do příslušného sektoru (adresou je přímo číslo sektoru), u SDHC se data adresují takzvanou blokovou adresou, která je vždy v násobcích čísla 512. Převod mezi bytovou a blokovou adresou je implementován velmi jednoduše, jedná se pouze o bitový posun 9× vlevo. U SDHC karet je zavedeno označení minimálních rychlostí podle tříd. Třída odpovídá minimální rychlosti zápisu v MB (class 2–10), čtení bývá zpravidla mnohem rychlejší.

SDXC (Secure Digital eXtended Capacity) se od SDHC liší technologií, takže maximální kapacita stoupla na 2 TB. Používá se zde stejně jako u SDHC bloková adresa a rychlosti jsou rozlišené třídami.

### 4.8.1 ENDIANITA

Endianita v informatice definuje způsob, v jakém pořadí se uloží byty číselného datového typu. V zásadě existují tři typy endianity, a to little-endian, middle-endian a big-endian. Karty většinou používají systém souborů FAT, který pracuje s pořadím bytů little-endian. Endianitu je nutné brát v úvahu hlavně při čtení a zápisu dat na kartu. Specifikace je taková, že na místo s nejnižší adresou se uloží nejméně významný byte (LSB) a za něj se ukládají byty až po nejvíce významný (MSB). Zjednodušeně by se dalo říct, že s čísly se pracuje „pozpátku“.

Jako dobrý příklad může posloužit třeba část oblasti FSInfo na kartě (viz níže), kdy jednotlivé čtveřice bytů vyjadřují 32-bitová čísla 0x00000000, 0x61417272, 0x000759EE a 0x00000026.

Offset (h)	00 01 02 03	04 05 06 07	08 09 0A 0B	0C 0D 0E 0F
000003E0	00 00 00 00	72 72 41 61	EE 59 07 00	26 00 00 00

Problematika SD paměťových karet je dále velmi podrobně rozebrána v bakalářské práci autora [26], kdy vznikl vlastní ovladač pro platformu Atmel XMEGA. V této práci jsou použity HAL knihovny od ST Microelectronics [22] pro přístup ke kartě, souborový je řízen knihovnamí FatFs [27], nyní oficiálně podporovanými výrobcem mikrokontroléru a softwarem STM32CubeMX [21].

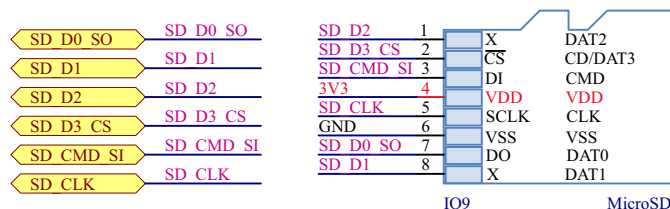
Knihovna FatFs a HAL knihovny (kap. 6) podporují všechny výše uvedené typy karet (SDSC, SDHC, SDXC).

### 4.8.2 KOMUNIKACE S KARTOU

Každá SD karta musí být podle specifikace [25] vybavena třemi typy komunikačních rozhraní. Je to SPI a dále 1-bit a 4-bit SD sběrnice. Zvolený protokol je vybrán při inicializaci zasláním příslušných příkazů, čímž se piny nastaví na příjem nebo odesílání dat. SD sběrnice ve 4-bit módu podporuje rychlosti až do 100 MB/s a zpřístupňuje některé speciální funkce jako ochranu dat (DRM) nebo další registry. Volbou SPI je možné dosáhnout přenosové rychlosti maximálně v řádu megabitů za sekundu.

Hardware LCD řadiče je navržen tak, aby bylo možné používat jak SPI, tak SDIO. Volba vždy záleží na vývojáři firmware. Princip zapojení je takový, že je karta na desce připojena k oběma sběrnicím a inicializuje se vždy jedna z nich (viz obr. 23). Piny druhé sběrnice musí zůstat ve stavu vysoké impedance.

V demonstračních aplikacích přiložených k této práci je využita nejrychlejší varianta SDIO 4-bit. Ovladač karty je součástí BSP (Board Support Package) knihoven a je úzce navázán na HAL vrstvu. Komunikace je možná standardním blokujícím módem (polling mode), ale i pomocí DMA (Direct Memory Access).



Obrázek 23: Paměťová karta

Název	MCU	Vlastnosti
SDIO_D[3..0]	PC[11..8]	SDIO Data
SDIO_CK	PC12	SDIO Clock
SDIO_CMD	PD2	SDIO Command

Tabulka 8: Připojení SD pomocí SDIO

Název	MCU	Vlastnosti
SD_MOSI	PG14	SPI6_MOSI
SD_MISO	PG12	SPI6_MISO
SD_CLK	PG13	SPI6_SCK
SD_CS	PC11	GPIO_Output

Tabulka 9: Připojení SD pomocí SPI

### 4.8.3 ORGANIZACE DAT NA KARTĚ

Základní paměťovou jednotkou (micro)SD karty je jeden sektor. Ten má většinou velikost 512 B a zároveň jde o nejmenší blok dat, který se dá na kartu zapsat, nebo z ní přečíst. Pokud je například potřeba změnit pouhý jeden bit na kartě, musí se přečíst celý sektor, jeden bit se změní a opět se celý sektor zapíše. To přináší vysokou přidanou režii (overhead) při práci s malými daty.

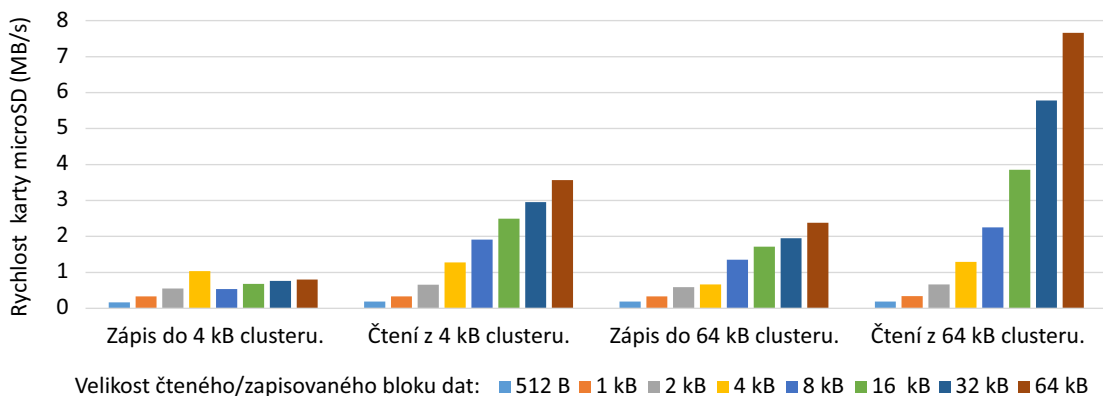
V souborovém systému FAT32 [24] se dále sektory spojují do clusterů, kdy jeden cluster je několiknásobek sektoru. Nejčastější velikost clusteru je 4 – 32 kB a tento parametr je možné zvolit při formátování. Karta přistupuje k datům pomocí příkazů, které umožňují zároveň číst/zapísat jeden sektor (CMD17/18 – READ/WRITE\_SINGLE\_BLOCK), pro blokový přístup je ale vhodné využít příkazy CMD24/25 (READ/WRITE\_MULTIPLE\_BLOCK). Použití příkazů je v této diplomové práci v režii implementovaného ovladače souborového systému FatFS [27]. Podrobný rozbor práce se souborovým systémem a paměťovou kartou je v bakalářské práci autora [26].

#### 4.8.4 TEST RYCHLOSTI KARTY

Velikost clusteru naformátované karty ovlivňuje rychlost zápisu a čtení. Nejvyšších hodnot karta zpravidla dosahuje při práci s bloky dat, které jsou stejně velké jako jeden cluster. Test proběhl dvakrát, karta byla naformátována postupně na 4 kB a 64 kB velikost clusteru. Metodika testování a rozbor zdrojového kódu je popsána v kap. 6.3.2. Výsledky měření jsou patrné z tab. (10) a obr. (24) níže.

blok dat	4 kB cluster		64 kB cluster	
	zápis (kB/s)	čtení (kB/s)	zápis (kB/s)	čtení (kB/s)
512 B	163	188	186	189
1 kB	332	333	330	337
2 kB	550	660	591	667
4 kB	1036	1280	668	1292
8 kB	538	1909	1356	2250
16 kB	680	2497	1719	3855
32 kB	762	2957	1952	5779
64 kB	803	3569	2377	7665

Tabulka 10: Test rychlosti paměťové karty microSD



Obrázek 24: Test rychlosti paměťové karty microSD

Výsledky ukazují, že se poměrně podstatně liší rychlost čtení a zápisu. Čtení bloku dat dosahovalo až několikanásobně vyšších hodnot než zápis stejného bloku, což je v praxi poměrně běžné a předem očekávatelné. Z měření je dále zřejmé, jak velikost clusteru naformátované karty ovlivňuje rychlost zápisu. Při velikosti clusteru 4 kB rychlost zápisu testovaného bloku rostla přímo úměrně s velikostí bloku zapisovaných dat, až do velikosti 4 kB (tzn. velikost zapisovaných blok byla stejná, jako velikost clusteru). Zvětšování zapisovaného bloku (sekvenční zápis více dat) pak dále nepřinesl zlepšení. Tuto skutečnost ilustruje první ze čtyř sad sloupců na obr. (24). Čtení vykazovalo přímou úměru s velikostí čtených dat. Nezanedbatelné zrychlení čtení přineslo přeformátování karty z 4 kB clusteru na 64 kB cluster (druhá vs. čtvrtá sada sloupců na obr. 24).

Přístup byl testován blokujícím módem na kartě Kingston microSDHC Class 10. Při využití DMA je pravděpodobné další zrychlení, takový test však z časových důvodů nebyl proveden.

## 4.9 PAMĚŤ PRO OBRAZOVÁ DATA

Mikrokontrolér STM32F439BIT6U je vybaven 256kB paměti RAM. To je naprosto postačující velikost pro běžný běh uživatelského programu, ale pro použití paměti jako bufferu pro displej je to nevyhovující. Mikrokontrolér je kvůli rozšíření kapacity naštěstí vybaven periferií FMC (Flexible Memory Controller), což umožňuje připojit několik typů paměti, mezi něž patří například Compact Flash, SDRAM, SRAM, PSRAM a NOR či NAND Flash [4].

Pro použití paměti jako bufferu pro displej je rozhodující vysoká rychlost a dobrý poměr cena ku kapacitě. Flash paměti bohužel příliš vysokou rychlostí nedisponují a mají omezený počet zápisů, takže pro tento účel nejsou vhodné. V úvahu přichází tedy jenom některá z typu RAM. Bohužel, paměti PSRAM jsou hůře sehnatelné a SRAM nemají dobrý poměr cena ku kapacitě, takže byla vybrána technologie SDRAM.

SDRAM (Synchronous Dynamic Random Acces Memory) je paměť typu DRAM se synchronním způsobem přenosu. Stejně jako ostatní varianty RAM, patří mezi volatilní paměti. To znamená, že se uložená informace bez přítomnosti napájení nenávratně smaže. Data je možno zapisovat nebo vyčítat s každým hodinovým tikem (Single Data Rate), proto se často používá označení SDR SDRAM. Nástupce těchto pamětí dokáží totiž číst a zapisovat na vzestupnou i sestupnou hranu hodinového tiku a proto se označují Double Data Rate (DDR) SDRAM.

Velikost paměti byla volena s ohledem na maximální podporované rozlišení, které je  $1024 \times 768$  pixelů (omezení řadiče), přičemž každý pixel je možné definovat maximálně 32-bitovou hloubkou ARGB8888 [17]. Tomu odpovídá maximální velikost jedné vrstvy bufferu pro displej (frame bufferu)

$$C = 1024 \cdot 768 \cdot 32 \text{ (bitů)} = 3 \text{ MiB.} \quad (20)$$

Na desku byl proto osazen paměťový modul MT48LC4M16A2 [6] od firmy Micron o velikosti 64 Mb (8 MiB). Při maximálním rozlišení je tedy možné s tímto modulem alokovat dvě vrstvy obrazu s hloubkou ARGB8888. Modul má šířku sběrnice 16 bit a je pinově kompatibilní se všemi 16-bit variantami v pouzdře TSOP (54 pin) až do velikosti 512 Mb [7]. Detaily pouzdra viz příloha (obr. 68).

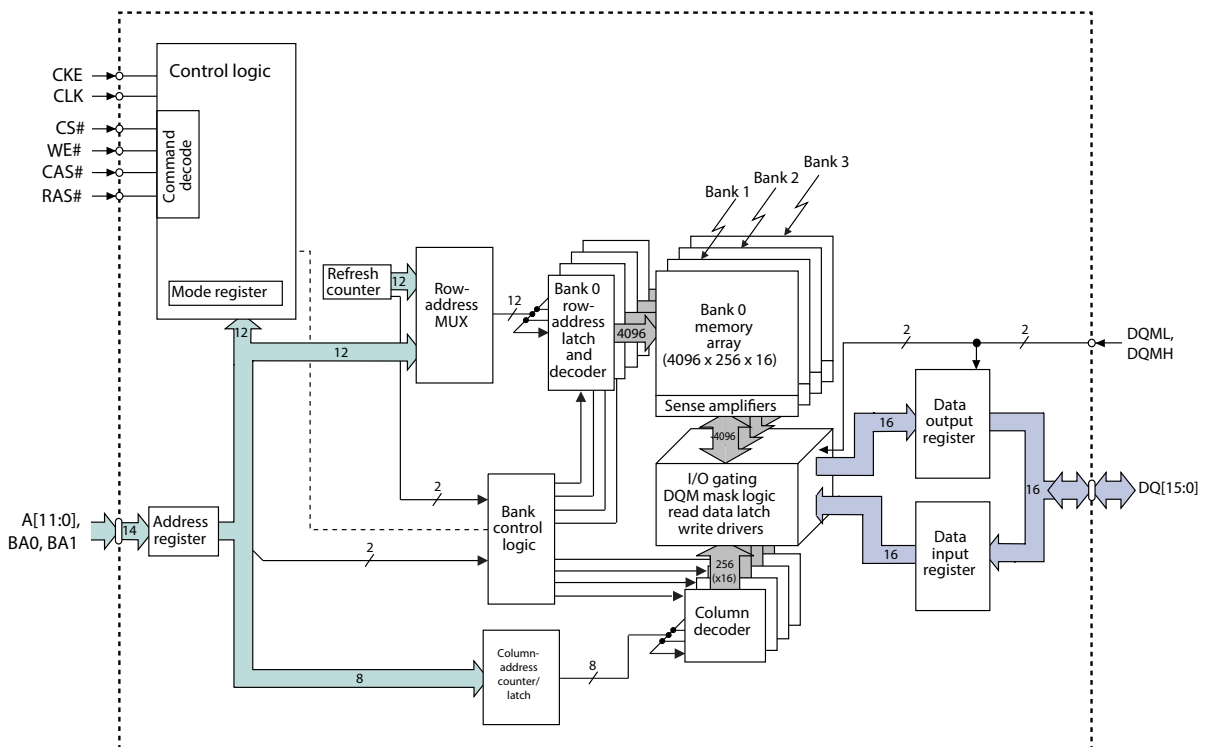
U mikrokontroléru byly záměrně nevyužity datové piny [31..16], které jsou připraveny pro případnou snadnou modifikaci desky plošných spojů pro šířku datové sběrnice 32 bitů a větší pouzdro TSSOPII (86 pin).

#### 4.9.1 VNITŘNÍ STRUKTURA SDRAM

Paměť je vnitřně obecně dělena na řádky, sloupce a banky. Použitý modul MT48LC4M16A2 je adresován 4096 řádky (12 bit), 256 sloupci (8 bit) a obsahuje 4 banky, šířka sběrnice je 16 bitů. Celková kapacita paměti v bitech je tedy

$$C = 4096 \cdot 256 \cdot 4 \cdot 16 = 67\,108\,864 \text{ (bit)} = 64 \text{ (Mb)} = 8 \text{ (MiB)} \quad (21)$$

Vnitřní struktura je patrná z obr. (25), obrázek byl převzat od společnosti Micron [6].



Obrázek 25: Vnitřní struktura SDRAM

Hlavní rozdíl mezi SRAM a SDRAM spočívá v tom, že SRAM má nedestruktivní čtení (informace se po přečtení buňky nesmaže) a komplikovanější strukturu paměťové buňky (SRAM MOS buňka se skládá ze šesti tranzistorů). SDRAM MOS buňka je složena pouze z jednoho tranzistoru a kondenzátoru, takže je podstatně levnější na výrobu. Čtení je ale destruktivní (kondenzátor se vybije) a buňky je třeba stále obnovovat (a přednabíjet). K tomu je přizpůsoben řadič paměti FMC.

Rychlost automatického obnovení řádků se nastavuje v registru SDRTR (SDRAM Refresh Timer Register), případně pomocí HAL knihoven funkcí `HAL_SDRAM_ProgramRefreshRate(&shander, hodnota)`.

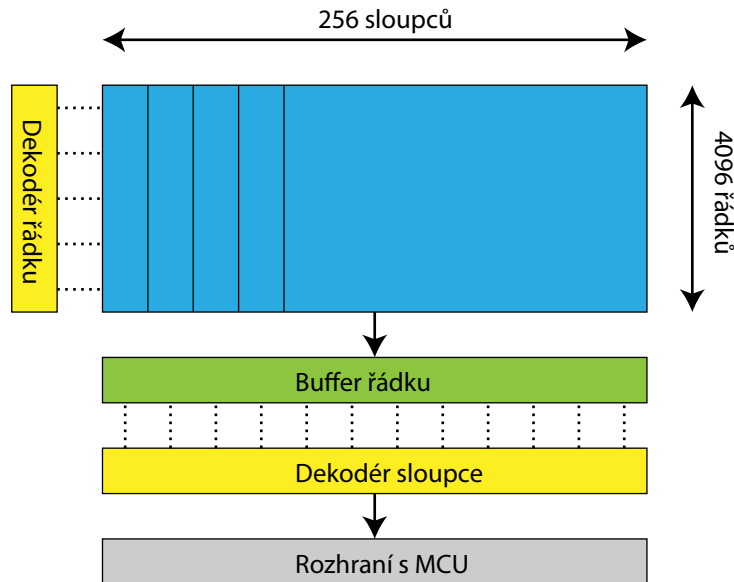
Hodnota registru závisí na frekvenci hodinového signálu CLK. Výrobce doporučuje, že se všechny řádky (celkem 4096) musí pravidelně nabíjet každých 64 ms. Parametr *hodnota* bude tedy

$$\frac{64 \text{ ms}}{4096} \cdot 84 \text{ MHz} - 20 = 1292 \quad (\text{tiků}) \quad (22)$$

Je doporučeno zachovat bezpečnou rezervu 20 hodinových tiků, než se požadavek dostane k SDRAM (například pokud se zrovna čte).

#### 4.9.2 PRINCIP ČTENÍ A ČASOVÁNÍ

Čtení probíhá postupně tak, že je nutné vybrat řádek, se kterým se bude pracovat (příkaz aktivace viz tab. 13). Potom následuje výběr čtení, nebo zápisu. Řádek je v paměti načten do bufferu a pomocí dekodéru je pak možné vyčíst konkrétní paměťovou buňku. Princip práce s pamětí zobrazuje obr. (26).



Obrázek 26: Princip čtení z paměti

Čas potřebný k adresaci řádků, sloupců a přechody mezi stavy paměti uvádí výrobce v nanosekundách, případně v násobcích hodinové frekvence. Parametry je třeba nastavit do periferie FMC ještě před inicializací čipu.

- $t^{CL}$  (CAS, latence). Počet hodinových cyklů, které uplynou mezi odesláním příkazu pro výběr sloupce a připravených datech na výstupu. Tedy počet tiků, které jsou třeba k přečtení bitu, pokud už je inicializován řádek.
- $t^{RCD}$ . Počet hodinových cyklů potřebných k vybraní řádku a přístupu do sloupce.
- $t^{RP}$ . Počet cyklů potřebných mezi zavoláním příkazu „precharge“ (tab. 13) a otevřením dalšího řádku.
- $t^{RAS}$ . Počet cyklů mezi stavem, kdy je řádek aktivní a znovu se přednabije.
- $t^{WR}$ . Počet cyklů, které musí uběhnout mezi posledním příkazem zápisu do řádku a přednabitím řádku.
- $t^{MRD}$ . Počet cyklů, než bude možné přistoupit k „Mode Registru“ (viz obr. 27).
- $t^{XSR}$ . Počet cyklů, který trvá přechod ze stavu „refresh“.
- $t^{RC}$ . Perioda, mezi aktivací dvou řádků.

### 4.9.3 PROCES INICIALIZACE SDRAM

SDRAM je nutné po zapnutí inicializovat a nastavit její parametry. To je provedeno v několika krocích.

- Nastavení časování periferie FMC. Hodnoty pro paměť MT48LC4M16A2 [6] jsou vypsány do tabulky tab. (11). Tabulka obsahuje doporučené hodnoty výrobce, řadič ale akceptuje násobky hodinové frekvence, na které je nutné časy přepočítat. Periferie FMC běží na polovině frekvence  $SYSCLK$ , takže jedna perioda hodinového signálu je

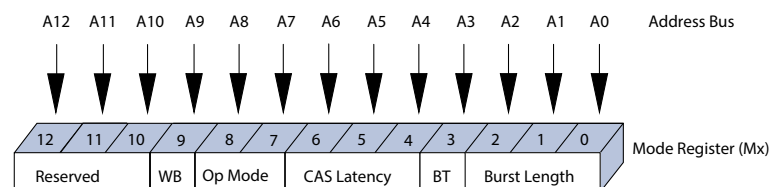
$$t_{CLK} = \frac{2}{SYSCLK} = \frac{2}{168 \text{ MHz}} = 11,9 \text{ (ns)} \quad (23)$$

K parametrům se dá přistupovat pomocí knihoven HAL [22] přes strukturu `FMC_SDRAM_TimingTypeDef`.

Parametr	Struktura HAL	Doporučená hodnota	Hodnota
$t_{MRD}$	LoadToActiveDelay	2 hodinové cykly	2
$t_{XSR}$	ExitSelfRefreshDelay	min. 70 ns	7
$t_{RAS}$	SelfRefreshTime	min. 42 ns	4
$t_{RC}$	RowCycleDelay	min. 70 ns	7
$t_{WR}$	WriteRecoveryTime	min. 1 cyk.+ 7 ns	2
$t_{RP}$	RPDelay	20 ns	2
$t_{RCD}$	RCDDelay	20 ns	2

Tabulka 11: Časování pro SDRAM periferie FMC

- Inicializace pinů paměti, hodin, přerušení a DMA. DMA se nastaví do režimu „Memory to Memory“, s automatickou inkrementací adresy a čtením/zápisem po slovech (32 bit word).
- Přivedení signálu CLK k SDRAM. Potom je nutné počkat alespoň 100 *us*.
- Vykonání příkazu Precharge All (tab. 13).
- Vykonání příkazu Auto Refresh (tab. 13).
- Naprogramování hodnot do „Mode Registru“ (obr. 27).
- Nastavení automatického obnovování paměťových buněk periferií FMC (vzorec. 22).



Obrázek 27: Mode Register paměti SDRAM

#### 4.9.4 ZAPOJENÍ SDRAM

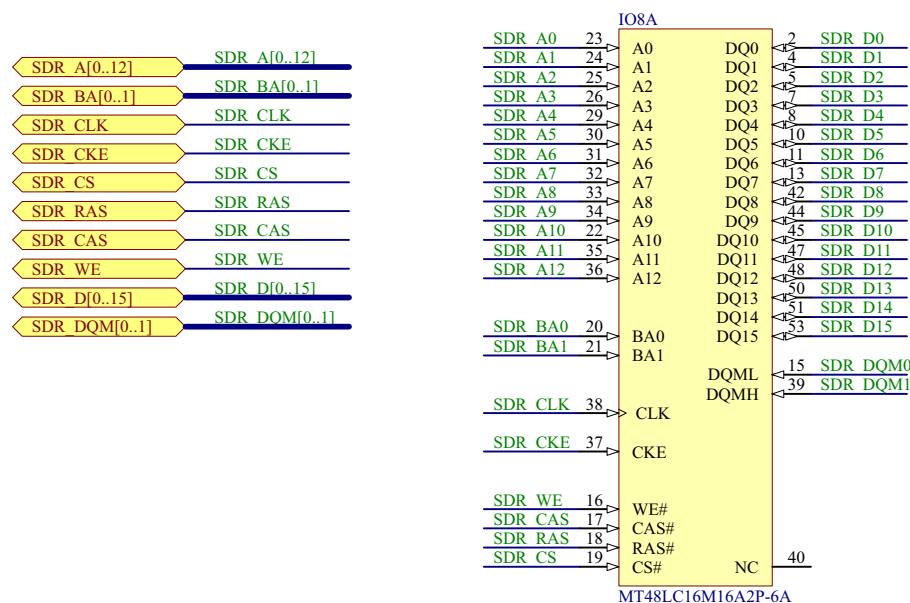
Paměť SDRAM se připojuje pomocí adresních, datových a řídicích vodičů, které jsou přímo součástí periferie FMC (Flexible Memory Controller). FMC umožňuje připojit až 2 paměťové čipy SDRAM, které využívají 2 interní banky FMC 5 a 6, které odpovídají SDRAM bankám 1 a 2.

- FMC banka 5 (= SDRAM banka 1), rozsah adres 0xC000 0000 - 0xCFFF FFFF
- FMC banka 6 (= SDRAM banka 2), rozsah adres 0xD000 0000 - 0xDFFF FFFF

Každá banka umožňuje adresovat až 256 MiB, takže maximální kapacita externí SDRAM mikrokontroléru je až 512 MiB. Této kapacity je možné dosáhnout pouze s použitím pouzdra TSSOPII 86 pin, či BGA pouzdra. K tomu by ale byla nutná úprava desky.

Velikost připojeného modulu v této práci je 8 MiB (64 Mbit) s využitím čipu MT48LC4M16A2 [6] od firmy Micron. S pouzdrem TSSOPII 54 pin je možné osazení i větší kapacity až velikosti 512 Mbit (64 MB), deska plošných spojů je tomu uzpůsobena a mezi moduly je zajištěna vzájemná pinová kompatibilita (příloha obr. 68). Větší kapacity než 512 Mbit (tedy až 512 MiB, jak bylo výše zmíněno) se dá dosáhnout pouze použitím pouzdra TSSOPII 86 pin.

Na obrázku (28) je ilustrováno zapojení pro 512 Mbit čip se 13 adresními vodiči A[12..0] (kvůli možnému rozšíření), použitá paměť MT48LC4M16A2 (64 Mbit) používá ale pouze 12 adresních vodičů A[11..0].



Obrázek 28: Paměť SDRAM

Periferie FMC umožňuje plně konfigurovat SDRAM rozhraní. Maximální rychlost je polovina, nebo třetina frekvence sběrnice AHB ( $HCLK/2$  či  $HCLK/3$ ). Ve všech kódech přiložených k této práci je využita rychlost  $HCLK/2$ , tedy 84 MHz.

V programu se s pamětí pracuje stejně, jako s jakoukoliv jinou částí RAM, adresování je od hodnoty 0xC000 0000. Je možné i namapování modulu přímo pro běh samotného kódu („heap“ a „stack“).



Připojení vodičů použitých ve schématu (obr. 28) k periférii FMC je zobrazeno v tabulce (12).

FMC	SDRAM	MCU	Význam
FMC_SDCKE0	CKE	PH2	Zapnutí hodinového signálu
FMC_SDNE0	CS	PH3	Výběr čipu
FMC_SDNRAS	RAS	PF11	Výběr řádku
FMC_SDNCAS	CAS	PG15	Výběr sloupce
FMC_SDNWE	WE	PC0	Zápis, nebo čtení
FMC_SDCLK	CLK	PG8	Synchronní hodinový signál
FMC_D[15..13]	DQ[15..13]	PD[10..8]	Data (16 bit)
FMC_D[12..4]	DQ[12..4]	PE[15..7]	
FMC_D[3,2]	DQ[3,2]	PD[1,0]	
FMC_D[1,0]	DQ[1,0]	PD[15,14]	
FMC_A[11,10]	A[11,10]	PG[1,0]	Adresa (max. 12 bit)
FMC_A[9..6]	A[9..6]	PF[15..12]	
FMC_A[5..0]	A[5..0]	PF[5..0]	
FMC_NBL[1..0]	DQM[1,0]	PE[1,0]	Horních/dolních 16 bitů
FMC_BA[1..0]	BA[1,0]	PG[5,4]	Číslo banky (1–4)

Tabulka 12: Mapování vodičů periférie SDRAM

Pro správnou funkčnost je nutné paměť na začátku programu inicializovat (kap. 4.9.3), kdy se zároveň do „Mode Registru“ SDRAM nakonfigurují konkrétní parametry, jako latence CAS, nebo automatické obnovování náboje paměťových buněk.

SDRAM je složená z řádků, sloupců a bank, ke kterým se přistupuje pomocí řídicích vodičů. Při čtení nebo zápisu musí nejprve proběhnout výběr řádku adresními vodiči, poté výběr sloupce. Tento proces je popsán v kap. 4.9.2 a obr. (26). Čtení, zápis, přednabití a obnovování paměťových buněk podléhají časování, stanoveného výrobcem (více popsáno v kap. 4.9.2).

Přepínání jednotlivých stavů je implementováno příkazy, které se provádí nastavením určité kombinace logických hodnot na řídicí a adresní piny. Příkazy obsahuje tabulka 13.

Příkaz	CS	RAS	CAS	WE	BA <sub>n</sub>	A10	A <sub>n</sub>
Ignorace příkazů.	H	x	x	x	x	x	x
Klidový stav (NOP).	L	H	H	H	x	x	x
Ukončení dávkového čtení, nebo zápisu.	L	H	H	L	x	x	x
Čtení z aktivního řádku.	L	H	L	H	banka	L	sloupec
Čtení s automatickým přednabitím řádku.	L	H	L	H	banka	H	sloupec
Zápis do aktivního řádku.	L	H	L	L	banka	L	sloupec
Zápis s automatickým přednabitím řádku.	L	H	L	L	banka	H	sloupec
Aktivace. Inicializace řádku pro čtení a zápis.	L	L	H	H	banka	řádek	
Přednabití. Zavření řádku vybrané banky.	L	L	H	L	banka	L	x
Přednabití všech řádků, všech bank.	L	L	H	L	x	H	x
Auto refresh. Obnovení řádku v každé bance.	L	L	L	H	x	x	x
Load Mode registr.	L	L	L	L	0	0	data

Tabulka 13: Příkazy pro paměť SDRAM

#### 4.9.5 TEST RYCHLOSTI SDRAM

Pro ověření správné funkce BSP ovladače (Board Support Package, kap. 6.2) SDRAM paměti byl proveden test rychlosti zápisu a čtení. Cílem testu bylo zjistit, jaký je rychlostní rozdíl mezi běžnými přístupy k paměti pomocí:

- Blokujícího módu. Přesunutí dat z interní SRAM mikrokontroléru do externí SDRAM pomocí standardních funkcí HAL knihovny (*BSP\_SDRAM\_Read/WriteData*).
- DMA převodníku. Přesun dat z SRAM do SDRAM funkcemi *BSP\_SDRAM\_Read/WriteData\_DMA*.

První část testu má tedy simulovat klasické metody, používané při práci s pamětí v mikrokontrolérech řady STM32F4xxx. Výsledky shrnují tab. (14) a obr. (29a). V rámci testu se zapisoval a četl 4 MB velký blok dat, metodika je podrobněji rozebrána v kap. 6.3.4. Je patrné, že s využitím DMA je rychlost znatelně vyšší a výhodou je taktéž skutečnost, že DMA nezatěžuje samotné Cortex jádro.

Druhá část testu sestává z využití DMA2D (Chrom-ART Accelerator) při zápisu do SDRAM. DMA2D je speciální periferie optimalizovaná pro práci s grafikou (grafický akcelerátor) a je součástí mikrokontrolérů řad STM32F4x7/9 (kap. 3.7 obr. 11).

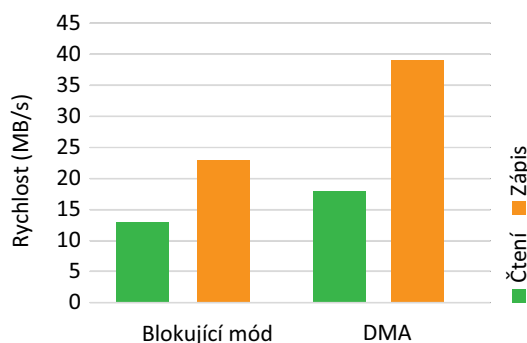
Níže, viz tab. (15) a obr. (29b), jsou zobrazeny dosažené rychlosti s DMA2D. Testování probíhalo tak, že byl 100× přeepsán frame buffer SDRAM odpovídající jedné obrazovce ( $800 \times 480 \times 4 \text{ B} \approx 1,5 \text{ MB}$ ). Zajímavý je fakt, že při vypnuté LCD obrazovce je dosažena téměř maximální teoretická rychlost paměti, v porovnání s případem zapnuté obrazovky. Grafický akcelerátor DMA2D tedy přináší  $2 \times - 4 \times$  vyšší rychlost oproti standardnímu DMA a  $5 \times - 9 \times$  násobné zrychlení vzhledem ke kopírování dat Cortex jádrem znak po znaku (obr. 29a a 29b nemají stejné měřítko).

	Čtení (MB/s)	Zápis (MB/s)
Blokující mód	13	23
DMA	18	39

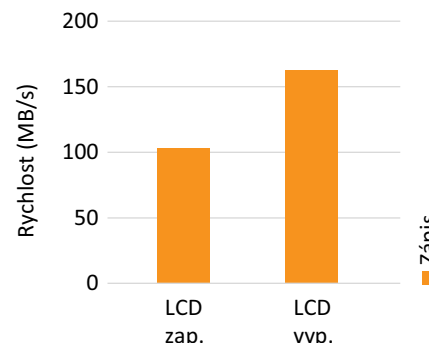
Tabulka 14: Standardní rychlost a DMA

LCD	Zápis (MB/s)
vyp.	163
zap.	101

Tabulka 15: DMA2D



(a) Rychlost přístupu do SDRAM



(b) Rychlost s pomocí DMA2D

Obrázek 29: Test rychlosti přístupu do SDRAM blokujícím módem, DMA a DMA2D

Při programování aplikace je tedy velmi výhodné si uvědomit přínos grafického akcelerátoru. Pokud je potřeba v paměti pracovat s obrázky, nebo je nutné vyplnit určitou plochu jednodílnou barvou, použití DMA2D podstatně zrychluje vykreslení, oproti standardním metodám známým z běžných řad mikrokontrolérů (blokující mód, nebo DMA).

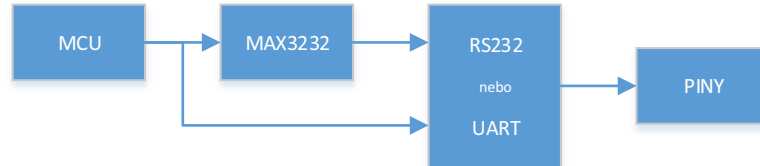
## 4.10 IMPLEMENTOVANÁ KOMUNIKAČNÍ ROZHRAŇÍ

### 4.10.1 RS-232

Požadavek zadání byl takový, aby bylo možné deskou komunikovat s okolím pomocí sériové linky. Procesor je vybaven několika periferiemi UART/USART (Universal Synchronou/Asynchronous Receiver Transmitter), které nejsou průmyslově nijak standardizované (kromě normy ISO 7816 pro Smart-Card), ale naprosto běžně se používají v praxi u drtivé většiny embedded zařízení. Je možné zvolit synchronní či asynchronní přenos, komunikační rychlost, délku jednoho bytu, jeho paritu a stop bit(y). Data se posílají duplexně nebo poloduplexně, nejméně významný datový bit je vyslán jako první.

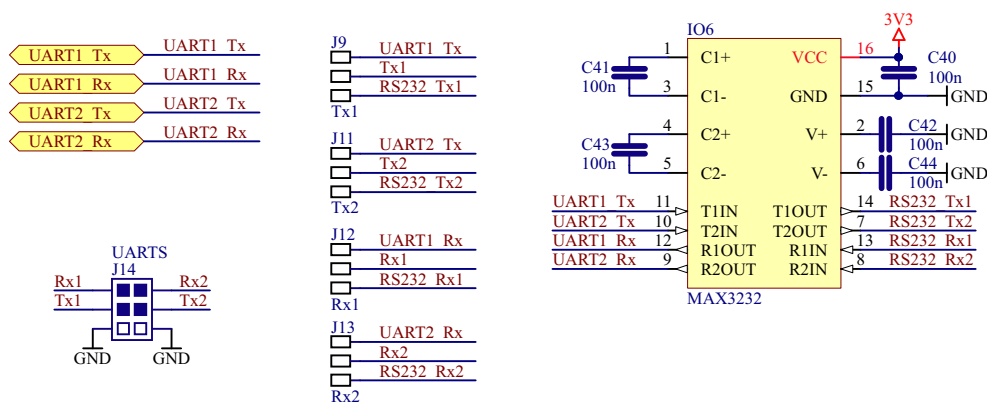
Průmyslovým standardem pro sériovou komunikaci je RS-232 (nebo jeho modifikace RS-422 a RS485). Průběh komunikace je v zásadě stejný jako u UART, sběrnice se ale liší použitým napětím (UART 3,3 V/5 V; RS-232  $\pm 12V$ ). Existují ale i varianty RS-232 s napětovými úrovněmi  $\pm 5 V$ ,  $\pm 10 V$  nebo  $\pm 15 V$ , nejsou však úplně běžné. RS-232 má standardizované navíc nepovinné vodiče pro řízení přenosu, ty však nejsou u tohoto LCD řadiče podporovány.

Pro kompatibilitu desky s UART 3,3 V napětovými úrovněmi a zároveň RS-232 sériovou linkou byl do schématu implementován čip MAX3232 [9] pro překlad. Úrovně je možné volit pomocí pájecích propojek, kdy se střední ploška spojí s jednou, nebo druhou krajní. Na obrázku obr. (30) je patrný způsob zapojení, kdy UART linka mikrokontroléru vstupuje zároveň do pájecích propojek i do MAX3232. Výstup se volí právě vhodným propojením buď přímo s mikrokontrolérem, nebo překladačem úrovní.



Obrázek 30: Blokové schéma převodníku UART/RS232

Elektrické zapojení je poměrně jednoduché. Z mikrokontroléru jsou využity periferie UART1 a UART2. MAX3232 je vybaven pouze několika externími kondenzátory pro nábojovou pumpu. V následujícím schématu (obr. 31) jsou kondenzátory C41 a C43 zapojeny jako plovoucí a C42, C44 a C40 slouží pro filtraci napětí.



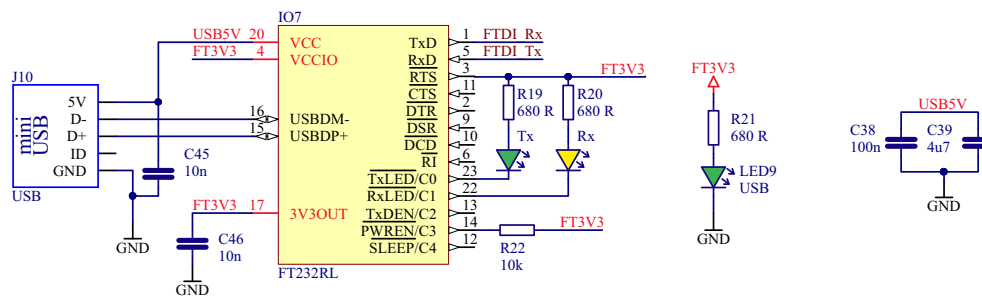
Obrázek 31: Převodník mezi UART a RS232

#### 4.10.2 FTDI

Konektivita s počítačem je zajištěna implementací obvodu FTDI od skotské firmy Future Technology Devices International (FTDI), pojmenovaný právě po firmě. Jedná se o ASIC (application-specific integrated circuit) standalone čip pro překlad UART sériové linky procesoru na USB protokol. Firma poskytuje ovladače a podporu pro všechny rozšířené operační systémy [15], které se nainstalují automaticky po připojení k počítači a vytvoří v operačním systému nový virtuální COM port (Communication port). Pro komunikaci se pak používá libovolný terminálový program.

Existuje i konkurence, která je znatelně levnější, ale nemá tak kvalitní podporu a ovladače se musí instalovat zvlášť, což není pro běžného uživatele příliš pohodlné. Například některé verze obvodu Prolific PL2303 mají problémy s instalací na novější operační systémy Windows 8 a vyšší (jedná se o varianty PL-2303HXA and PL-2303X) a firma Prolific do budoucna nezvažuje ani jejich podporu a implementaci. Ukončena byla i podpora pro Windows XP, který je stále v embedded zařízeních rozšířený.

Zapojení je poměrně jednoduché, viz obr. (32). Z počítače jsou připojeny pouze linky USB data+ a data- a napájení 5 V pro samotný FTDI čip. Ten má vlastní 3V3 stabilizátor, ze kterého dokáže napájet indikační LED diody pro připojení (USB) a komunikaci (Rx, Tx). Napětí je filtrováno několika kondenzátory. Napětí 3V3 nebo 5V jsou voleny pinem *VCCIO*.



Obrázek 32: Připojení k USB pomocí FTDI čipu

Ze strany procesoru je FTDI připojeno na piny 10 (*Tx*) a 11 (*Rx*) portu *B* tab. (16). Ve zdrojovém kódu jsou piny definovány makry pro preprocesor v hlavičkovém souboru desky, spolu s makry pro RS-232 linky. To přináší mnoho výhod, protože BSP ovladač (Board Support Package) pro desku řadiče stačí pouze přiložit ke zdrojovému kódu aplikace a zavolat funkci *BSP\_COM\_Init()* s parametry *ftdi* nebo *COM1* či *COM2* pro RS-232 linky. Podporována je komunikace v blokujícím režimu (polling mode), ale i přerušování a DMA (Direct Memory Access). Defaultní rychlost komunikace je baud 115 200, 8 bit, bez parity, 1 stop bit.

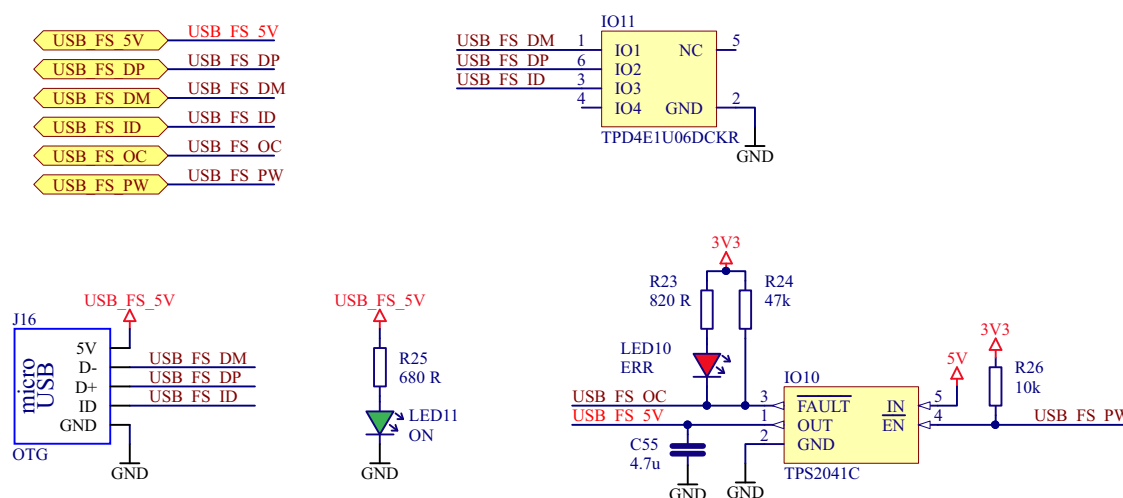
#define BOARD_FTDI	USART3
#define BOARD_FTDI_TX_PIN	GPIO_PIN_10
#define BOARD_FTDI_TX_GPIO_PORT	GPIOB
#define BOARD_FTDI_TX_AF	GPIO_AF7_USART3
#define BOARD_FTDI_RX_PIN	GPIO_PIN_11
#define BOARD_FTDI_RX_GPIO_PORT	GPIOB
#define BOARD_FTDI_RX_AF	GPIO_AF7_USART3
#define BOARD_FTDI_IRQn	USART3_IRQn

Tabulka 16: Připojení čipu FTDI k mikrokontroléru

### 4.10.3 USB ON-THE-GO FULLSPEED

Komunikační rozhraní s počítačem doplňuje USB On-The-Go Fullspeed. Hardwarově je podporován režim „Device“ i „Host“. Je tak možné desku řadiče připojit například jako virtuální COM port k počítači (místo čipu FTDI, kap. 4.10.2). Naopak, příkladem implementace režimu „host“ je třeba zprovoznění USB flash disku pro rozšíření kapacity desky řadiče a snadnější přenos uživatelských dat.

Elektrické zapojení je na obrázku (33). Mechanicky co nejbližže microUSB AB konektoru je umístěn supresor napětí TPD4E1U06. Slouží k ochraně desky před elektrickými špičkami, způsobenými statickou elektřinou. Přepínání mezi jednotlivými režimy je vyřešeno čipem TSP2041C od Texas Instruments. Jedná se o elektronický přepínač s přepětovou ochranou, přímo určen pro USB aplikace.



Obrázek 33: USB On-The-Go Fullspeed

Mikrokontrolér rozhoduje, který režim (device/host) bude použit a podle toho řídí, zda do sběrnice *USB\_FS\_5V* bude dodávat napětí, nebo je napětí získáno externě. K přepínání slouží pin *USB\_FS\_PW* (power). Elektronický spínač obsahuje i proudovou ochranu, která je signalizována rozsvícením červené LED a vodičem *USB\_FS\_OC* (overcurrent). Přehled zbývajících pinů a jejich funkcí na mikrokontroléru obsahuje tab. (17).

Název	Pin	Vlastnosti
USB_FS_5V	PA9	USB_OTG_FS_VBUS
USB_FS_DP	PA12	USB_OTG_FS_DP
USB_FS_DM	PA11	USB_OTG_FS_DM
USB_FS_ID	PA10	USB_OTG_FS_ID
USB_FS_OC	PG3	GPIO_Input
USB_FS_PW	PD11	GPIO_Output

Tabulka 17: Mapování pinů USB OTG Fullspeed

Bohužel, kvůli nedostatku času se nepodařilo softwarově implementovat a správně odladit USB OTG část desky. Hardwarová implementace by však měla být správná.

## 4.11 AUDIO

Moderní vestavěné zařízení kladou daleko větší požadavky na uživatelské rozhraní, než tomu bylo před několika lety. Nejenom z hlediska grafiky a jednoduchosti, ale i co se týče odezvy. Proto vznikl požadavek vybavit TFT LCD řadič audio výstupem. Je tak možné například jednoduše signalizovat uživateli nečekané situace a akusticky potvrzovat vstupy.

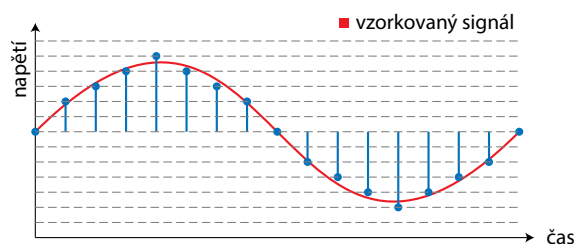
Běžně používané metody pro převod digitálního audio signálu na analogový ve vestavěných zařízeních jsou například:

- Pulsně šířková modulace (PWM, obr. 34). Informace je přenášena dvouhodnotovou veličinou, nejčastěji napětím či proudem. Signál má konstantní frekvenci, ale různou střídu, pomocí níž je modulován přenášený výkon v rozsahu 0–100%. PWM je demodulována vhodnou analogovou dolnofrekvenční propustí. Tento typ používají velmi jednoduchá zařízení, na která nejsou velké požadavky na přesnost a kvalitu zvuku – například různá pípání a osmibitové zvuky. Při dobré implementaci je možné dosáhnout ale i kvalitnějšího výstupu.



Obrázek 34: PWM modulace 25 %

- D/A převodník. Digitální audio data jsou přivedena na D/A převodník, který je přímo převede na analogovou hodnotu. Potom signál nejčastěji ještě vyhlazuje dolnofrekvenční propustí. Data bývají kódována pomocí pulzně kódové modulace (PCM, obr. 35) a jako převodník může posloužit periferie mikrokontroléru, externí D/A, R-2R rezistorová síť, nebo delta-sigma ( $\Delta\Sigma$ ). Je takto možné dosáhnout velmi kvalitního výstupu, což ale často vyžaduje větší množství externích součástek.



Obrázek 35: PCM modulace

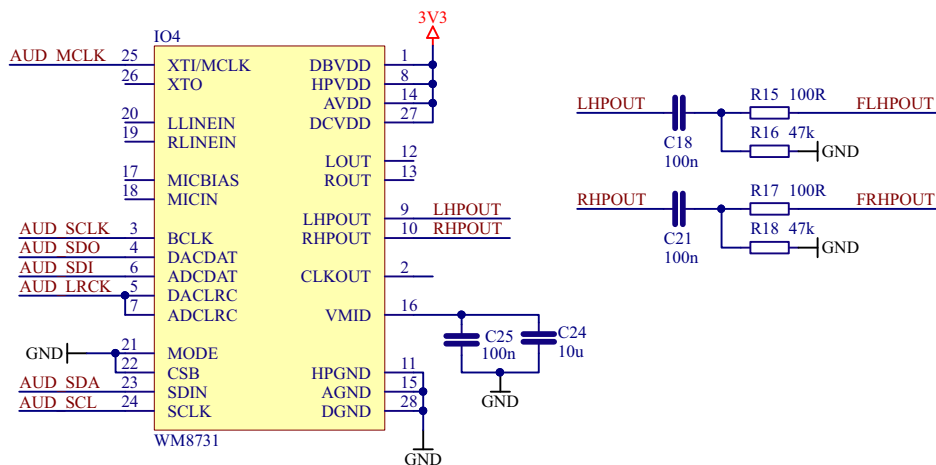
- Audiokodek. V hardwarovém slova smyslu se jedná o externí součástku, na jejíž vstup jsou přivedena digitální data, modulovaná pomocí PCM. Používá se synchronní sběrnice, nejčastěji SPI či I2S (kap. 4.11.2). Data jsou kodekem interpolována, převedena pomocí interních D/A převodníků a filtrována na odstranění vyšších harmonických. Často je umožněna i opačná funkce, kdy je kodekem vzorkován a kvantován externí signál (např. mikrofonní vstup), který je takto převeden do digitální podoby. Mezi nejznámější audiokodeky patří například Realtek HD Audio Codec používaný ve zvukových kartách počítačů, či řada WM8xxx od firmy Wolfson používaná i v přehrávačích iPod. Řadič popisovaný v této práci používá audiokodek Wolfson WM8731 [11] (kap. 4.11.1).

#### 4.11.1 AUDIOKODEK WM8731

Audiokodek slouží k převodu digitálních audio dat na analogová. Data jsou uvnitř interpolována a převáděna pomocí interních A/D a D/A převodníků. Byl vybrán kodek WM8731 [11] britské společnosti Wolfson Microelectronics plc. Čip je osazen například v zařízení iPod 3G a nabízí velmi kvalitní sluchátkový a linkový audio výstup, možné je i připojení mikrofonu. Mikrofon ani linkový vstup zde nejsou použity. Konstrukce audio výstupu je navíc velmi levná, celé řešení se pohybuje okolo 100 Kč.

Podporované jsou vzorkovací frekvence 8 kHz, 32 kHz, 44.1 kHz, 48 kHz, 88.2 kHz a 96 kHz. Velikost slova může být 16, 24 a 32 bitů, ovšem interní A/D a D/A sigma-delta převodníky jsou 24 bitové. Data jsou směrem ven i dovnitř posílána po sběrnici I2S (kap. 4.11.2, tab. 19), využívá se PCM modulace (obr. 35). Vnitřní registry se nastavují pomocí sběrnice I2C (tab. 18). Bohužel je možný pouze zápis, nikoliv čtení registrů. Adresa na sběrnici je 0x32 s vodičem CSB připojeným na zem.

Elektrické zapojení je na obr. (36). Ke kodeku je možné připojit linkový vstup a výstup (L/RLINEIN, L/ROUT), toho ale není využito. Signál je dále do zesilovače (kap. 4.11.4) přiveden pomocí sluchátkového výstupu L/RHPOUT, který je možné softwarově vypnout („mute“). Linkový výstup toto neumožňuje. Kodek nabízí i připojení mikrofonu a odesílání navzorkovaných dat do mikrokontroléru po I2S (na schématu vodič SDI/ADCDAT), mikrofon ale není implementován.



Obrázek 36: Audiokodek WM8731

Následující tabulky (tab. 18 a 19) obsahují přehled použitých pinů a jejich namapování na mikrokontrolér.

Pin	MCU	Název
PB8	I2C1_SCL	AUD_SCL
PB9	I2C1_SDA	AUD_SDA

Tabulka 18: Mapování vodičů I2C

Pin	MCU	Název
PC6	I2S2_MCK	AUD_MCLK
PD3	I2S2_CK	AUD_SCLK
PC2	I2S2_ext_SD	AUD_SDI
PC3	I2S2_SD	AUD_SDO
PB12	I2S2_WS	AUD_LRCK

Tabulka 19: Mapování vodičů I2S

#### 4.11.2 SBĚRNICE I2S

Sběrnice I2S slouží k digitální komunikaci mezi audiokodekem a mikrokontrolérem. Data jsou přenášena PCM modulací (obr. 35). Standard byl vyvinut firmou Philips (nyní NXP) v roce 1986 a obsahoval 3 vodiče. *SCK* (synchronous clock) pro hodinový signál, *SD* (serial data) pro navzorkovaná data a *WS* (word select) jako indikátor kanálu.

V dnešní době se používají další vodiče, které firma Wolfson pojmenovává jako *MCLK* (master clock), který má frekvenci obvykle 256 násobku vzorkovací frekvence (*BLCK*, Philips *SCK*) a data z mikrofonu (u Wolfson *ADCDAT*). Firma ST Microelectronics používá vlastní značení podobné původní specifikaci Philips, pro data z mikrofonu je použit název *I2S\_ext\_SD*. Přehled názvů vodičů je v tabulce (20).

Aby nedošlo k nedorozumění mezi značením výrobců a původní specifikací, bylo ve schématech (obr. 14 a obr. 36) zavedeno vlastní značení.

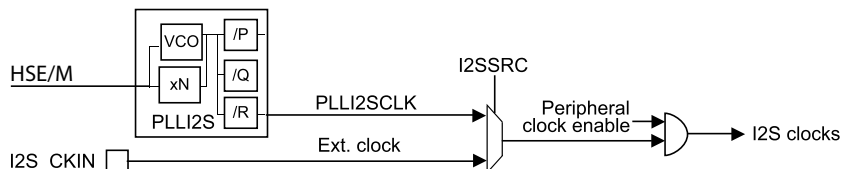
- *MCLK* („master clock“). Hlavní hodiny pro kodek, 256 násobek vzorkovací frekvence.
- *SCLK* („synchronous clock“). Reference pro PCM modulovaná audio data.
- *SDO* („serial data out“). Z pohledu mikrokontroléru výstup dat.
- *SDI* („serial data in“). Z pohledu mikrokontroléru vstup dat.
- *LRCK* („left/right clock“). Výběr levého či pravého kanálu.

Význam	Philips	WS	STM	Schéma
Master clock, 256x WS	X	MCLK	I2S_MCK	AUD_MCLK
Základní clock	SCK	BCLK	I2S_CK	AUD_SCLK
Audio data z MCU do audiokodeku	SD	DACDAT	I2S_SD	AUD_SDO
Výběr kanálu	WS	DACLRC	I2S_WS	AUD_LRCK
Výběr kanálu	WS	ADCLRC	I2S_WS	AUD_LRCK
Audio data z mikrofonu do MCU	X	ADCDAT	IS_ext_SD	AUD_SDI

Tabulka 20: Pojmenování vodičů na sběrnici I2S

Referenční signál pro sběrnici I2S je získán PLL násobičkou (obr. 37) dosazením do struktury *PLLI2S* podle vzorce (24).

$$f_{I2S} = \frac{F_{HSE}}{PLL_M} \cdot \frac{PLLI2S_N}{PLLI2S_R} \quad (24)$$

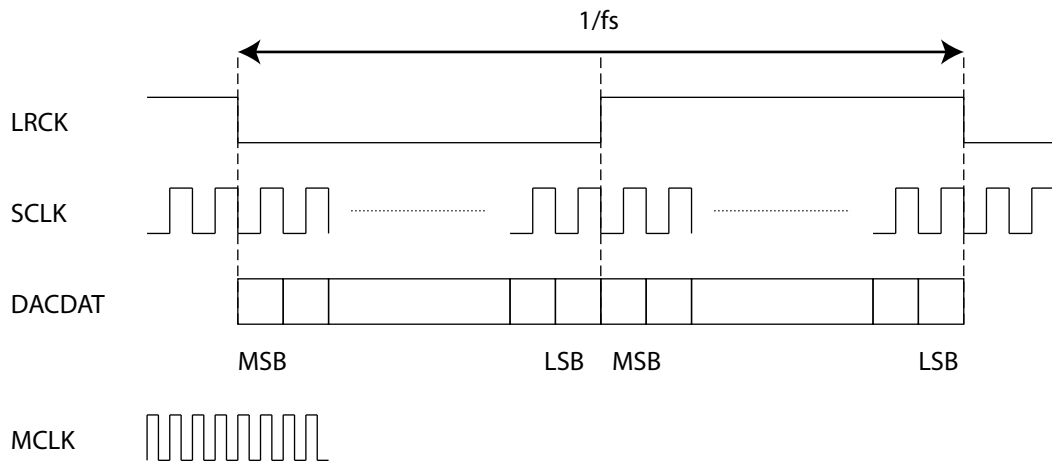


Obrázek 37: Časování sběrnice I2S



### 4.11.3 PŘÍKLAD NASTAVENÍ AUDIOKODEKU

Kodek umožňuje nastavení různých frekvencí, počtu vzorků, zarovnání bitů a polaritu „word select“ signálu (*LRCK*). Jako model nastavení může posloužit například CD formát s vzorkovací frekvencí 44,1 kHz a velikostí vzorku 16 bitů. Přehled časování je zřejmý z obr. (38).



Obrázek 38: Průběh signálu na sběrnici I2S

Signál *LRCK* (výběr levého či pravého kanálu) bude mít tedy frekvenci 44,1 kHz, takže perioda je

$$\frac{1}{f_s} = \frac{1}{44100} = 22,68 \quad (\mu s). \quad (25)$$

Od něj se odvíjí frekvence *MCLK* (reference pro kodek kvůli interpolaci a decimaci), která bude

$$256 \cdot 44,1 \text{ kHz} = 11,29 \quad (\text{MHz}). \quad (26)$$

Čip nabízí nastavit i 384, či 250/272 násobek vzorkovací frekvence, například při použití USB jako reference. To je provedeno zápisem příslušných hodnot do registru R8 „Sampling Control“.

Při délce vzorku 16 bitů a dvou kanálech bude mít hodinový signál *SCLK* pro data frekvenci

$$44,1 \cdot 16 \cdot 2 = 1,411 \quad (\text{MHz}), \quad (27)$$

přičemž velikost vzorku se nastavuje v registru R7 na bitech IWL[1..0].

Zápis do registrů je realizován posláním dvou bytů na I2C adresu 0x32, přičemž horních 7 bitů obsahuje fyzickou adresu registru, nejnižší bit prvního bytu a celý druhý byt nesou informaci o vlastní hodnotě.

S použitím HAL knihoven může být zápis podobný následujícímu kódu.

```
buffer[0] = ((RegisterAddr<<1)&0xFE) | ((RegisterValue>>8)&0x01);
buffer[1] = RegisterValue&0xFF;
HAL_I2C_Master_Transmit(&handler, 0x32, buffer, velikost, timeout);
```

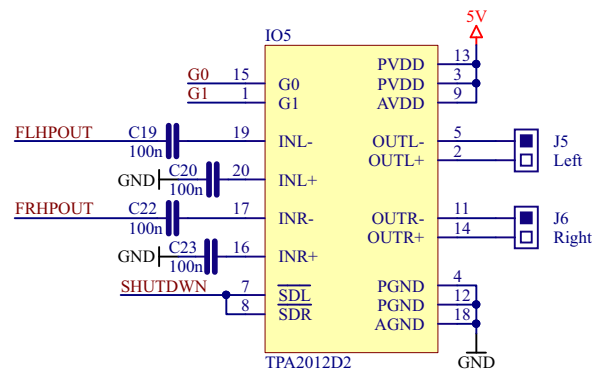
#### 4.11.4 ZESILOVAČ TPA2012

Sluchátkový výstup z audiokodeku (kap. 4.11.4) má velmi malý výkon, jímž by nebylo možné vybudit reproduktor. Proto bylo nutné implementovat ještě aktivní zesilovač. Audio zesilovače se dělí do tříd, kdy každá třída má své specifické vlastnosti a účinnost. Nejčastěji používané jsou:

- Třída A. Používají se tranzistory, které pracují v lineární části své charakteristiky. Účinnost je velmi malá, ale linearita je velmi vysoká.
- Třída B. Každá půlvlna je zesilována vlastním tranzistorem, na jeden kanál jsou tedy nutné dva prvky. Účinnost je mnohem vyšší než u třídy A, ale v oblasti, kde signál prochází nulou, může vzniknout velké zkreslení.
- Třída D. Zesilovač pracuje v pulsním režimu. Spínaný signál je získán pulsně-šířkovou modulací PWM (obr. 34), potom je využita dolní propust. Účinnost je velmi vysoká a celý zesilovač může být integrován i ve velmi malém pouzdře.

Pro účely diplomové práce byl zvolen zesilovač třídy D, konkrétně TPA2012 [8] od firmy Texas Instruments. Výkon pro pouzdro QFN je 2,1 W/kanál při napájení 5 V na 4  $\Omega$  reproduktoru, nebo 1,4 W/kanál na 8  $\Omega$  reproduktoru.

Elektrické schéma je patrné z obr. (39). Není potřeba mnoho externích součástek, pouze vstupní kondenzátory pro odstranění stejnosměrné složky. Napájení je odebíráno z 5 V větve ze spínaného zdroje (kap. 4.12).



Obrázek 39: Zesilovač TPA2012

K mikrokontroléru (kap. 4.2, obr. 14) jsou připojeny pouze 3 vodiče pro ovládání – *SHUTDWN* pro vypnutí a *G[1..0]* pro hrubé nastavení zesílení. Jejich mapování je patrné z tabulky (22), nastavení zesílení piny *G[1..0]* je v tabulce (21).

Zesílení	G0	G1
6 dB	0	0
12 dB	0	1
18 dB	1	0
24 dB	1	1

Tabulka 21: Zesílení

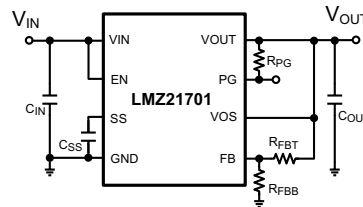
Název	Pin	Funkce
AUD_G0	PB4	GPIO_Output
AUD_G1	PB5	GPIO_Output
AUD_SHUT	PK7	GPIO_Output

Tabulka 22: Mapování vodičů zesilovače

## 4.12 NAPÁJENÍ

Napájení celé hlavní desky řadiče je zajištěno dvěma spínanými zdroji. Část s mikrokontrolérem, podsvícením LCD, audiokodekem, MAX3232, SDRAM a microSD kartou je napájena napětím 3.3V, USB OTG FS a audiozesilovač využívají 5V větve.

Schéma zapojení zdroje je patrné z obr. (40). Byly použity spínané zdroje LMZ21701 od firmy Texas Instruments [12]. Podporují vstupní napětí v rozmezí 3 - 17 V a výstupní proud 1 A. Účinnosti spínaných zdrojů jsou při běžném provozu přes 80 %, při nejvyšším zatížení můžou dosáhnout okolo 90% (obr. 41). Dodávají se v pouzdře 8 $\mu$ SIP (3.5 mm  $\times$  3.5 mm). Vyžadováno je jenom minimum okolních součástek. Vstupní a výstupní filtrační kondenzátor ( $C_{IN}$  a  $C_{OUT}$ ), softstart kondenzátor ( $C_{SS}$ ), napětový dělič ( $R_{FBT}$  a  $R_{FBB}$ ) a pull-up rezistor ( $R_{PG}$ ) pro open drain „Power Good“ indikační pin.



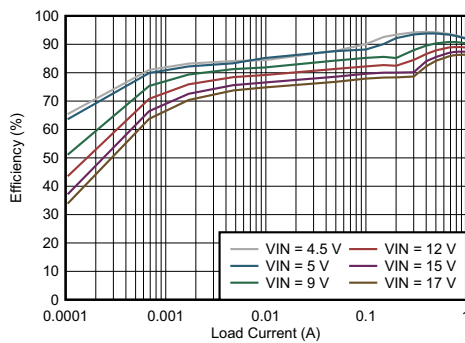
Obrázek 40: Spínané zdroje pro napájení desky a USB OTG

Soft-start je doporučeno nastavit na 0,5 ms a víc. Kondenzátor  $C_{SS}$  je nabíjen interním proudovým zdrojem 2,9  $\mu$ A. Doporučená kapacita je 3,3 nF, což při dosazení do vzorce (28) vychází zhruba 1,5 ms.

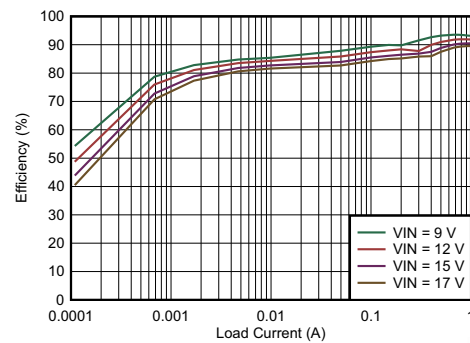
$$T_{SS} = C_{SS} \cdot \frac{1,25 \text{ V}}{2,9 \mu\text{A}} \quad (28)$$

Hodnoty zpětnovazebních odporů se dají vypočítat ze vzorce pro napětový dělič (29) vzhledem k interní referenci 0,8 V. Výrobce doporučuje používat  $R_{FBB}$  menší než 400 k $\Omega$  a zároveň stanovuje doporučené hodnoty pro 3,3 V výstup  $R_{FBT} = 1.21 \text{ M}\Omega$ ,  $R_{FBB} = 383 \text{ k}\Omega$  a pro 5V výstupní napětí  $R_{FBT} = 232 \text{ k}\Omega$ ,  $R_{FBB} = 44,2 \text{ k}\Omega$ .

$$R_{FBT} = R_{FBB} \cdot \frac{V_{OUT}}{0,8} - 1 \quad (29)$$



(a) Účinnost pro výstupní napětí 3V3



(b) Účinnost pro výstupní napětí 5V

Obrázek 41: Účinnosti měničů

## 4.13 GPIO PINY, LED A TLAČÍTKO

Požadavek zadání byl vybavit desku několika univerzálními digitálními vstupy a výstupy. Pro tento účel byla osazena standardní dvouřadá lišta („header“), na kterou je vyvedeno 8 GPIO pinů (general purpose input/output). Rozteč je 2,54 mm a GPIO doplňuje napájení 3V3 a GND. Přehled pinů obsahuje tabulka (23).

Název	Pin	Vlastnosti
GPIO0	PD5	Input, Output, Analog, EXTI5
GPIO1	PD4	Input, Output, Analog, EXTI4
GPIO2	PD7	Input, Output, Analog, EXTI7
GPIO3	PJ12	Input, Output, Analog, EXTI12
GPIO4	PA15	Input, Output, Analog, EXTI15
GPIO5	PG9	Input, Output, Analog, EXTI9
GPIO6	PG10	Input, Output, Analog, EXTI10
GPIO7	PG11	Input, Output, Analog, EXTI11

Tabulka 23: Mapování pinů GPIO

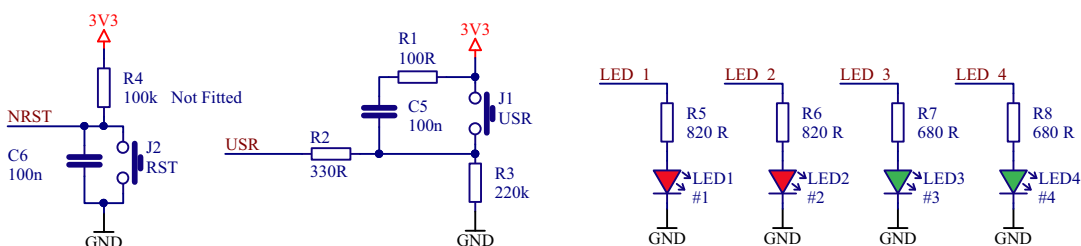
Deska je vybavena čtyřmi LED diodami pro signalizaci stavů, implementována jsou i dvě tlačítka. *RST* pro reset a *USR* pro všeobecné využití. Přehledy použitých pinů obsahují tab. (24 a 25) zapojení je zřejmé z obr. (42). Tlačítka jsou ošetřena proti zákmitům. Pin *PA0* tlačítka *USR* byl zvolen záměrně i kvůli funkci „SYS.WKUP“, takže je tak možné probudit mikrokontrolér z úsporného režimu, pokud do něj bude přepnut.

Název	Pin	Barva
LED1	PD12	Červená
LED2	PD13	Červená
LED3	PG6	Zelená
LED4	PG7	Zelená

Tabulka 24: Mapování LED

Název	Pin	Vlastnosti
USR	PA0	SYS.WKUP, EXTI10
RST	NRST	Reset

Tabulka 25: Mapování tlačítek na desce



Obrázek 42: Zapojení tlačítek a LED diod na desce řadiče

Softwarová podpora je obsažena v knihovně BSP (Board Support Package) přiložené k této práci. Knihovna obsahuje inicializační a obslužné funkce, LED a tlačítko jsou popsány pomocí maker pro preprocesor kompatibilní s HAL knihovnami (Hardware Abstraction Layer) od verze 1.5 [22]. Je tak možné zavolat inicializaci či obsluhu (například *LED1*) jednoduchými příkazy, jako:

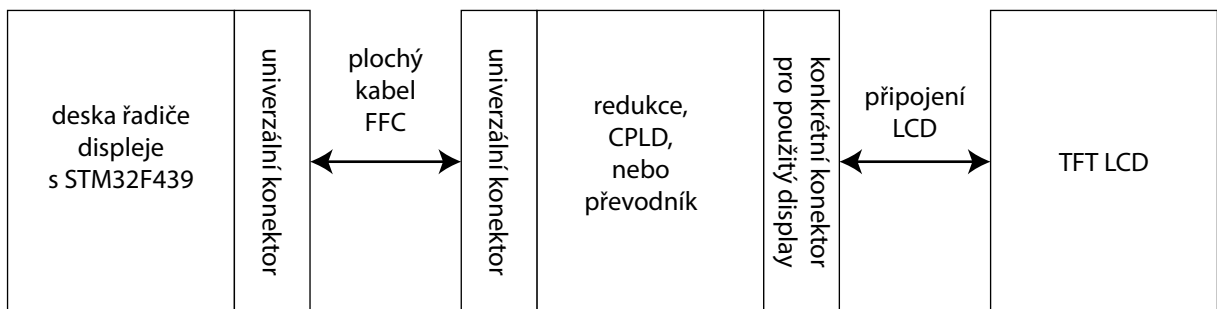
```
BSP_LED_Init(LED1);
BSP_LED_On(LED1);
```

## 5 VÝSTUP NA DISPLEJ

Bylo nutné vymyslet, jak bude navržen výstup z mikrokontroléru, aby bylo dosaženo co nejvyšší univerzálnosti a k řadiči se dal připojit libovolný displej. Na základě analýzy panelů od několika výrobců [14] bylo zjištěno, že konektory pro různé displeje nejsou nijak standardizované a každý má jinou velikost a jiný počet pinů, proto není možné obecně displej přímo připojovat k hlavní desce řadiče. Neexistuje žádný standard, který by definoval rozložení pinů na LCD konektorech, ani šířky sběrnice. Trendem ovšem zůstává, že se nejčastěji používají konektory s 35 - 50 piny a barevná sběrnice je paralelní, a to nejčastěji 16, 18, nebo 24 bitová. Rozložení vodičů má ale každý výrobce jiné.

Jako vhodné řešení se jeví výstup obrazových dat přímo z mikrokontroléru doplnit o několik dalších, často používaných signálů. Tyto signály budou společně s napájením přivedeny do univerzálního konektoru na desce řadiče. Univerzální konektor je dále popsán v kap. 5.1.

Filozofie použitého řešení ilustruje obr. (43). Výstup z univerzálního konektoru je přiveden do menší desky, která bude zajišťovat konverzi signálu pro každý konkrétní typ kompatibilního panelu. Výstup z desky konverze je přímo přiveden na LCD panel.



Obrázek 43: Připojení libovolného LCD displeje k hlavní desce

Konverze může probíhat několika způsoby. Možné varianty jsou:

- Pevná redukce pro konkrétní model LCD panelu. Takové řešení je velmi levné, zahrnuje pouze dva konektory, které se mezi sebou vhodně propojí. Pokud připojený displej obsahuje dotykovou vrstvu, na redukci pro to může být specializovaný řadič. Varianta pevné redukce byla v rámci této diplomové práce vyrobena pro použitý panel ET0700G0DH6 [5], více viz kap. 5.2. Popis použitého řadiče dotykové vrstvy je v kap. 4.7.
- Aktivní, programovatelná redukce s možným mapováním pinů. Výhodou takového konceptu je, že deska může podporovat více druhů panelů, pokud mají stejný konektor. Rozložení pinů se potom nastaví softwarově přeprogramováním. K tomu účelu byl navržen a vyroben modul s CPLD (complex programmable logic device). Řešení je dále rozebráno v kap. 5.3.
- Specializovaný převodník. Řadič se nemusí omezovat pouze na výstup pro paralelní sběrnice. Pomocí specializovaného převodníku je možné konvertovat digitální signál na analogový, čímž by bylo možné připojit se například k televizi, monitoru či projektoru. Varianta nebyla v rámci diplomové práce dále zkoumána, pro konverzi na VGA signál by se dal však poměrně jednoduše využít například převodník ADV7123 [30] umožňující až 10 bit/kanál digitální vstup a RGB analogový VGA výstup.

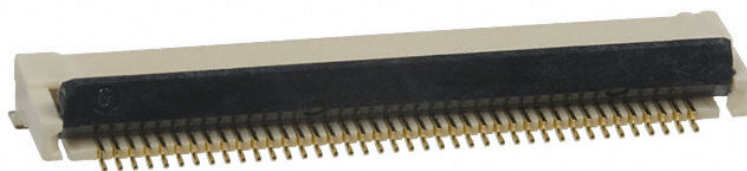
## 5.1 UNIVERZÁLNÍ KONEKTOR PRO TFT LCD DISPLEJ

Pro podporu napříč různými typy displejů byl navržen univerzální konektor pro TFT LCD. Obsahuje sběrnice Red (R), Green (G) a Blue (B) pro osmibitovou hloubku barev, doplněnou o standardní signály horizontální a vertikální synchronizace (*HS*, *VS*) a základní hodinový signál pro displej (*CK*).

Některé dotykové panely obsahují přímo řadič pro dotykovou vrstvu (touchscreen), který se připojuje pomocí SPI (*MOSI*, *MISO*, *SCK*, *SS*) nebo I<sup>2</sup>C (*SDA*, *SCL*), proto jsou i tyto sběrnice na konektoru přítomné. Mnoho řadičů pro dotykovou vrstvu umožňuje vyvolat přerušení (interrupt) při dotyku, implementován je tedy pin *INT*. Displeje je často možné vypínat a zapínat, proto je přítomen vodič *PWR*, případně *RST* pro reset interní logiky. Podsvícení se řídí nejčastěji analogově velikostí napětí, nebo digitálně pomocí pulsně-šířkové modulace (*PWM*, obr. 34). Možné jsou obě varianty díky pinům *PWM* a *DAC*. Pro jiné funkce (tlačítka, joystick nebo LED na displeji) je možné použít 5 GPIO (General-purpose input/output) pinů. Podrobný přehled periferie LTDC a použitých pinů je v tab. (26) a na obr. (45).

Pro připojení redukcí byl nakonec vybrán konektor společnosti Omron, konkrétně 50 pinová varianta XF2M-5015-1A. Rozteč pinů je 0,5 mm, připojovat je možné 50 pin plochý konektor typu FFC (Flexible Flat Cable) se stejnou roztečí. Kompatibilní ploché kabely vyrábí například firma Molex, nebo Würth Elektronik. Socket je typu ZIF (Zero Insertion Force), takže kabel je jistěn zámkem a vkládání je velmi snadné. Na propojování je možné využít kabel typu ZIF-50, který se v praxi používá na připojování miniaturních 1,8" pevných disků, nebo Compact Flash karet v některých embedded aplikacích a spotřební elektronice.

Na obrázku (44) je ukázka konektoru Omron typu XF2M.



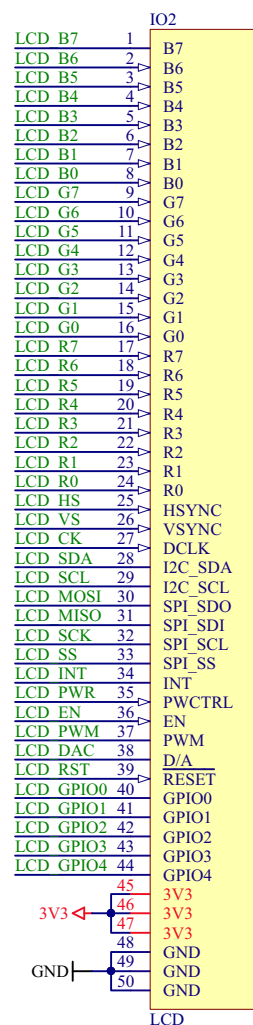
Obrázek 44: Konektor Omron XF2M-4015-1A

Napájení elektroniky displejů obvykle není energeticky moc náročné a vystačí si s napětím 3,3 V a proudem několik miliampér. Velkou spotřebu má ale podsvícení, což se pohybuje v řádu stovek mA, žádný ze zkoumaných panelů však nepřesáhl 1 A. Konektory XF2M a ploché kabely ZIF dokáží přenést maximálně 0,5 A na jedné lince, proto je i s rezervou univerzální konektor vybaven třemi napájecími linkami pro 3,3 V a GND.

Následující tabulka (tab. 26) obsahuje přehled namapovaných pinů procesoru na univerzální konektor pro LCD, z obrázku obr. (45) jsou zřejmé i napájecí linky.

LTDC	MCU	Význam
LTDC_R7	PJ6	Červený kanál (8 bit)
LTDC_R6	PB1	
LTDC_R[5..4]	PJ[4..3]	
LTDC_R3	PB0	
LTDC_R[2..1]	PJ[1..0]	
LTDC_R0	PI15	
LTDC_G[7..5]	PK[2..0]	Zelený kanál (8 bit)
LTDC_G[4..0]	PJ[11..7]	
LTDC_B[7..6]	PK[6..5]	Modrý kanál (8 bit)
LTDC_B5	PA3	
LTDC_B4	PK3	
LTDC_B[3..1]	PJ[15..13]	
LTDC_B0	PE4	
LTDC_HS	PI12	Časování
LTDC_VS	PI13	
LTDC_CLK	PI14	
LTDC_SDA	PH5	I2C
LTDC_SCL	PH4	
LTDC_MOSI	PA7	SPI
LTDC_MISO	PA6	
LTDC_SCK	PB3	
LTDC_SS	PJ2	
LTDC_INT	PB2	
LTDC_PWR	PB14	Zapnutí displeje
LTDC_EN	PF10	Signál EN
LTDC_PWM	PA1	PWM pro podsvícení
LTDC_DAC	PA5	D/A pro podsvícení
LTDC_RST	PB13	Reset
LTDC_GPIO[4..3]	PF[8..9]	Univerzální piny
LTDC_GPIO[2..1]	PH[7..6]	
LTDC_GPIO0	PJ5	

Tabulka 26: Mapování pinů periferie LTDC

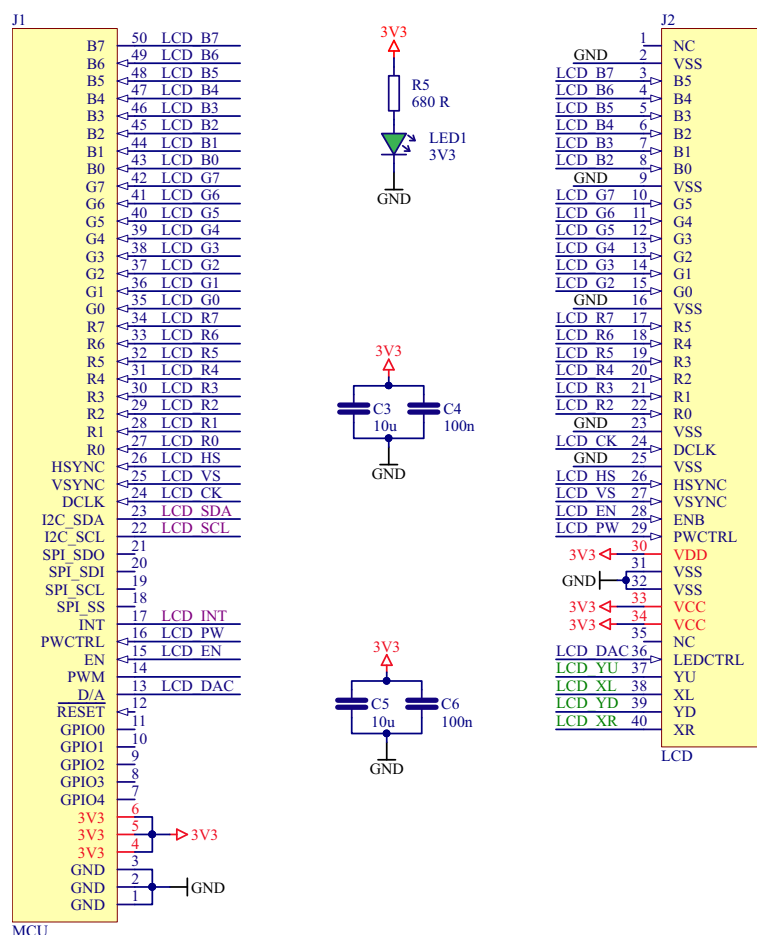


Obrázek 45: Univerzální konektor

## 5.2 REDUKCE PRO DISPLEJ ET0700G0DH6

Nejjednodušší varianta, jak připojit konkrétní displej k univerzálnímu konektoru (kap. 5.1), je výroba jednoduché pasivní redukce, která pouze propojuje potřebné vodiče z 50 pin varianty univerzálního konektoru (kap. 5.1) na plochý 40 pin konektor, kterým je vybaven použitý displej ET0700G0DH6 [5]. Panel ET0700G0DH6 implementuje šestibitové paralelní rozhraní na kanál, kvůli tomu je nutné konvertovat signál z osmibitového rozhraní univerzálního konektoru. Toho musí být docíleno tak, že jsou propojeny nejvyšší bity obou rozhraní, takže se linky RGB[7..2] 50 pin konektoru spojí s linkami RGB[5..0] displeje. Tímto způsobem jsou zanedbány nejnižší, nejméně důležité bity obrazové informace. Propojení ilustruje obr. 46.

Při konstrukci první verze redukce toto nebylo dodrženo a byly propojeny nejnižší bity obou rozhraní, takže linky RGB[5..0] 50 pin konektoru „intuitivně“ odpovídaly vodičům RGB[5..0] použitého displeje. Tím byla ignorována nejdůležitější část obrazové informace a redukce způsobovala problémy zobrazení, dále popsané a nasimulované v kapitole 3.3.



Obrázek 46: Schéma redukce pro displej ET0700G0DH6

Použitý displej obsahuje i rezistivní dotykovou vrstvu. Redukce je vybavena řadičem pro konverzi z analogových výstupů čtyřvodičového rozhraní (kap. 3.4) na I2C protokol. Řadič je dále popsán v kap. 4.7. Analogová část je připojena k vodičům *XU*, *XL*, *YD* a *XR*; digitální I2C výstup je připojen k pinům *LCD\_SCL* a *LCD\_SDA*. Příchozí přerušení vyvolané dotykem snímá pin *LCD\_INT*.

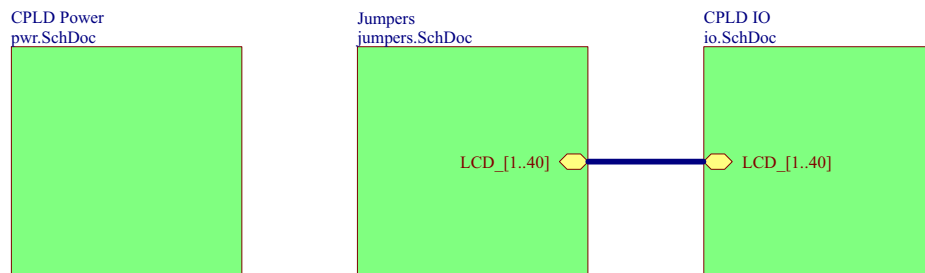


### 5.3 PROGRAMOVATELNÁ REDUKCE CPLD

Kvůli podpoře všech typů panelů s různými rozloženými pinů bylo nutné vymyslet způsob, jakým namapovat piny univerzálního konektoru (kap. 5.1) na konkrétní rozložení libovolného displeje. Pro tento účel byla vytvořena programovatelná redukce s obvodem CPLD (Complex Programmable Logic Device). Byl vybrán typ 5M80ZT100C5N z řady MAX V [9] od firmy Altera. Obvod je dodáván v pouzdře LQFP 100 a obsahuje 79 programovatelných I/O pinů.

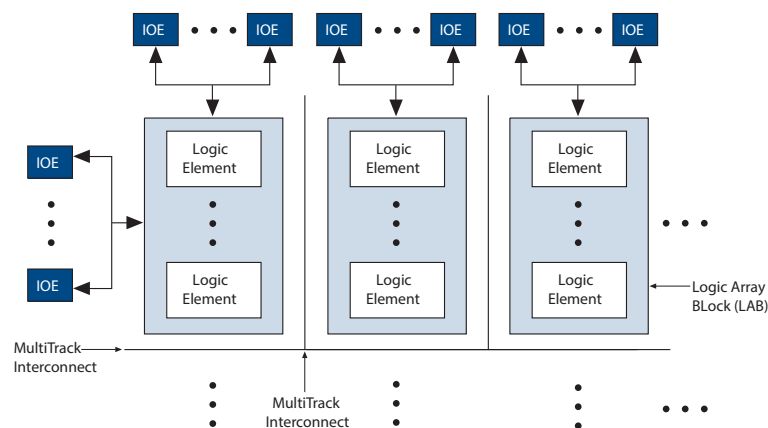
Redukce musí umožňovat mapování datových, ale i napájecích pinů pro libovolný displej, rozložení těchto pinů ale není nijak standardizováno. Napájecí proud může mít kvůli podsvícení spotřebu celkem až 1 A. Proudová zatížitelnost na pin CPLD je ale maximálně 25 mA. Tento problém je v redukci vyřešen tak, že se datové piny propojí přímo přes CPLD, kdy proud 25 mA postačuje. Napájecí piny se v CPLD nakonfigurují jako vstup s vysokou impedancí a napájecí napětí se do displeje přivede pomocí série pájecích propojek, kterými je redukce pro tento účel vybavena. Napětí přivedené pájecími propojkami takto piny CPLD nezníčí.

Blokové schéma zařízení je patrné z obrázku (obr. 47), které rozděluje redukci na tři části – napájecí část obsahuje stabilizátor (*power.SchDoc*), protože jádro CPLD vyžaduje napětí 1,8 V. Druhý blok zahrnuje sérii pájecích propojek (*jumpers.SchDoc*) a poslední část implementuje samotnou logiku a programovací mechanismus (*io.SchDoc*).



Obrázek 47: Schéma programovatelné redukce s CPLD

Každé CPLD je složeno z bloků logických polí *LAB* (Logic Array Blocks), které jsou uspořádány do řádků a sloupců. Řady 5M40Z – 5M240Z obsahují vždy 6 sloupců a 4 řádky, celkem tedy 24 *LAB*. Logické pole se dále dělí na logické elementy *LE* (Logic Element) a řídicí signály. Zdroj obrázku (obr. 48) je [3].

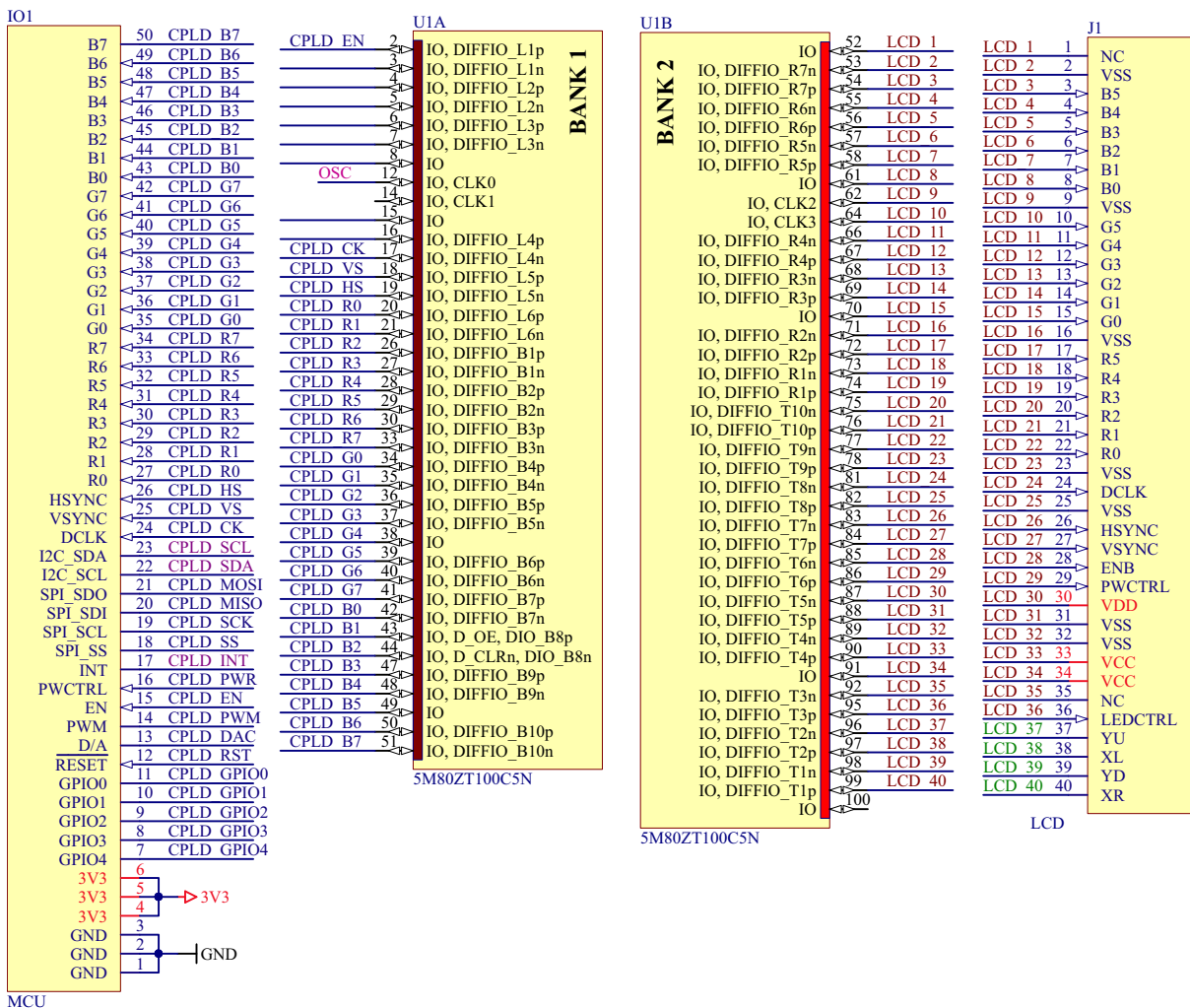


Obrázek 48: Vnitřní struktura CPLD

### 5.3.1 KONFIGURACE VSTUPŮ A VÝSTUPŮ

Použité programovatelné logické pole 5M80ZT100C5N se skládá ze dvou IO (Input-Output) bank. Každá banka může mít vlastní napájení, k oběma bylo ale připojeno napětí 3,3 V. Výhoda použití CPLD spočívá v tom, že se vodiče z 50 pin univerzálního konektoru (kap. 5.1) připojí přímo ke vstupům pole na Bance 1, aniž by se linky musely křížit. Vnitřní propojení logických elementů LE je naprogramováno v jazyce VHDL a výstupy jsou odvedeny z Banky 2 do 40 pin konektoru k LCD panelu, opět bez křížení.

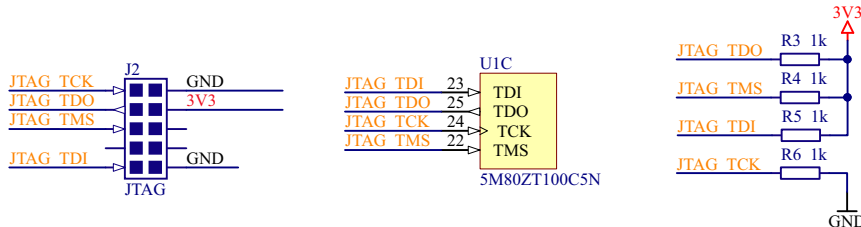
Schéma desky je na obr. (49).



Obrázek 49: Mapování vstupů a výstupů do CPLD

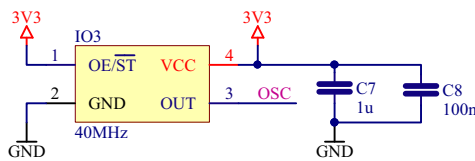
Omezením při implementaci tohoto řešení je skutečnost, že pole CPLD nedokáže přenést analogové signály. Není tedy možné programově mapovat linky z elektrod dotykové vrstvy. Ty jsou napevno vedeny z pinů LCD [40..37] do řadiče dotykové vrstvy (kap. 4.7). Digitální výstup z řadiče je připojen k I2C vodičům CPLD\_SCL a CPLD\_SDA, přerušení při dotyku signalizuje CPLD\_INT.

Pro programování logického pole CPLD bylo implementováno standardní rozhraní JTAG. K tomu jsou vyhrazeny speciální piny 22-25 pouzdra LQFP100. Využívají se pouze linky *TCK*, *TDO*, *TMS* a *TDI*, které jsou přivedeny do standardního 10 pin JTAG konektoru. Pro nahrání kódu je možné použít například programátor USB Blaster od firmy Terasic. Na základě doporučení výrobce byly použity i pull-up a pull-down rezistory. Programovací část ilustruje obr. (50).



Obrázek 50: CPLD programovací rozhraní JTAG

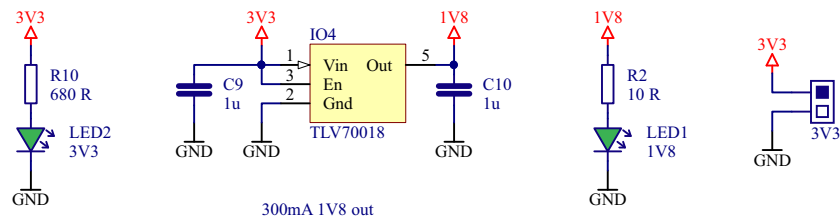
Referenční signál pro interní logiku CPLD je získáván z externího oscilátoru 40 MHz. Oscilátor ale není využit, protože by byl porušen vzorkovací teorém při frekvenci vzorkovaném signálu obrazových dat 16,125 MHz (kap. 4.5 obr. 18) a vzorkovacím signálu 40 MHz. Oscilátor nevyužívají ani připojené Schmidty klopné obvody na vstupech CPLD pro tvarování signálu. Zapojení je patrné z obr. (51).



Obrázek 51: CPLD oscilátor

Jak již bylo naznačeno v kapitole 5.3, každá z bank 1 a 2 programovatelného pole umožňuje používat jinou úroveň napětí pro logické vstupy a výstupy. Obě banky jsou napájeny napětím 3,3 V. Jádra obou bank vyžadují ale nižší napětí, takže bylo nutné implementovat ještě jednoduchý lineární stabilizátor pro 1,8 V. Ten je realizován miniaturním obvodem TLV70018 v pouzdře SOT23-5 od firmy Texas Instruments. Vstup i výstup jsou opatřeny signalizačními LED diodami. Výstupní dioda na napětí 1,8 V svítí velmi slabě, pro orientační zjištění funkčnosti stabilizátoru to ale postačuje.

Schéma napájecí části jádra je na obr. (52).



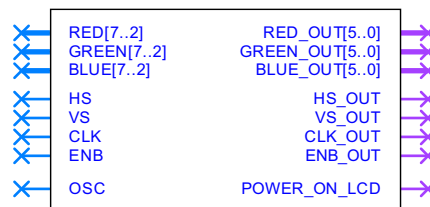
Obrázek 52: Napájení jádra CPLD

### 5.3.2 PROGRAMOVÁNÍ CPLD

Programování logického pole je možné například v jazyce VHDL (VHSIC Hardware Description Language). Jedná se o programovací jazyk, který slouží k popisu hardware. Pro vývoj je vhodné použít přímo vývojové prostředí Quartus, dodávané výrobcem Altera, v omezeně funkční verzi zdarma.

Základní jednotkou je entita, která popisuje propojení vstupů a výstupů aktivní redukce pro displej. S linkami obrazových dat RGB je vhodné pracovat jako s vektory. Je nutné propojit vždy horní (nejdůležitější) bity vektorů, jinak se obrazová informace deformuje (kap. 3.3). Vstupní piny 8 bit/kanál sběrnice RGB[7..2] tedy budou přivedeny k výstupním pinům 6 bit/kanál sběrnice RGB[5..0].

Redukce propojuje taky signály horizontální a vertikální synchronizace displeje *HS* a *VS* a vodič pro potvrzování platných řádků *ENB* (kap. 4.6, obr. 19). V řídicí desce je nutné vybrat mód, kterým bude displej ovládán (*HS+VS* vs. *ENB*). Entita implementuje i vstupní pin z externího oscilátoru *OSC* a výstupní vodič pro zapnutí podsvícení LCD. Struktura entity je patrná z obr. (53).



Obrázek 53: Struktura entity CPLD

Proces implementovaný do CPLD je poměrně triviální. Propojuje pouze vstupní piny RGB sběrnice a řídicí signály s výstupními. Podsvícení displeje je zapnuto trvale, protože do CPLD čipu nebyl přiveden příslušný vodič z mikrokontroléru.

Ukázka VHDL kódu:

```
RED_OUT<=RED;
GREEN_OUT<=GREEN;
BLUE_OUT<=BLUE;

HS_OUT<=HS;
VS_OUT<=VS;
CLK_OUT<=CLK;
ENB_OUT<=ENB;

POWER_ON_LCD<='1';
```

Práce s barevnými kanály a zahazování dolních bitů může vypadat následovně:

```
RED : in std_logic_vector(7 downto 2);
RED_OUT : out std_logic_vector(5 downto 0);
```

Obyčejné vstupy a výstupy jsou definovány jako:

```
HS : in std_logic;
HS_OUT : out std_logic;
```

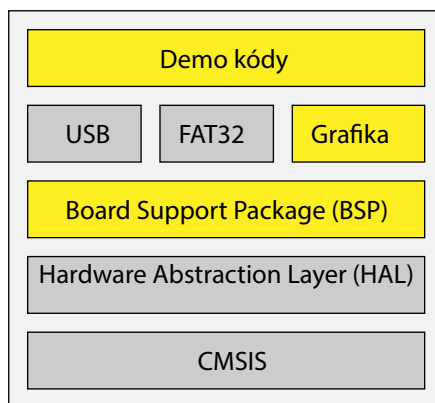
U vstupů a výstupů CPLD je vhodné nastavit parametry „Fast Input/Output register“ a „Weak Pull-Up Resistor“ na *zapnuto*.

## 6 SOFTWARE

Pro tvorbu software byly použity freeware nástroje, poskytované společností ST Microelectronics. Projekty byly založeny a nakonfigurovány pomocí software STM32CubeMX [21], kde je možné uživatelsky příjemným způsobem nastavit a inicializovat veškeré periferie a časování mikrokontroléru. Software nabízí generování do několika vývojových prostředí (Atollic, IAR, Keil a SW4STM). Každý z příložených projektů ve složce „*Projects*“ obsahuje vlastní konfigurační soubor s příponou „*.ioc*“ pro STM32CubeMX.

Právě SW4STM, tedy vývojové prostředí System Workbench for STM32 [32] je novinka roku 2015, která je vyvíjena komunitou nad Eclipse. Je multiplatformní a zdarma, ale jsou v ní ještě malé chyby a nedodělky, správně nefungoval například debug. Přesto bylo prostředí úspěšně použito při programování demo kódů v této práci. Pro upload firmware do mikrokontroléru byla využita utilita STLink [35].

Mapa zdrojového kódu je zobrazena na obr. (54). Kód se dělí do jednotlivých logických vrstev. Nejnižší úroveň je CMSIS (Cortex Microcontroller Software Interface Standard), kde je obsaženo standardizované API pro tvorbu aplikací nad Cortex jádrem. CMSIS obsahuje i definice adres registrů a maker pro preprocesor, knihovny pro DSP (Digital Signal Processing) a základní API pro RTOS (Real Time Operating System).



Obrázek 54: Mapa zdrojového kódu

V logické vrstvě nad CMSIS se nacházejí HAL knihovny (Hardware Abstraction Layer [22]). Jedná se opět o softwarové rozhraní, sjednocující programování různých periférií napříč různými Cortex jádrem a řadami mikrokontrolérů. HAL knihovny jsou od konce roku 2014 preferovány společností ST Microelectronics a doporučovány pro vývoj aplikací, proto bylo knihoven využito i v této práci.

Jeden z hlavních cílů byla implementace vlastních ovladačů pro použité periferie na navržené desce. Tyto ovladače je zvykem označovat zkratkou BSP (Board Support Package), využívají funkcí HAL od verze 1.5 (březen 2015, verze 1.5 obsahuje novou, nekompatibilní definici maker pro preprocesor). Jedná se především o obsluhu GPIO a LED (kap. 4.13), dotykovou vrstvu (kap. 4.7), LCD řadič (kap. 4.5), microSD kartu (kap. 4.8), paměť SDRAM (kap. 4.9.4) a audio (zesilovač kap. 4.11.4 + kodek kap. 4.11.1).

Pro otestování BSP knihoven bylo vytvořeno několik malých ukázkových aplikací. Ty využívají vlastní vygenerované fonty (kap. 6.1) a jejich podrobnější popis je v (kap. 6.2). Kvůli podpoře FAT32 [24] byla implementována knihovna FatFs [27].

## 6.1 PODPORA FONTŮ

LCD řadič bylo třeba vybavit podporou vlastních fontů, aby splňoval požadavky zadání. Font je v typografii definován jako kompletní sada znaků abecedy jedné velikosti a jednotného stylu. Písmo se dále dělí na rodiny, přičemž jedna rodina je skupina řezů, odvozená od jednoho písma. Obvykle se používá **tučný řez**, *kurzíva* a jiné. Mnoho fontů je možné najít online, například na webu dafont.com [37].

Nejčastěji používané fonty mívají různé šířky jednotlivých znaků (např. W a i), což zpřehledňuje čitelnost delšího textu. V programování, terminálových aplikacích a tabulkách se ale často používají rodiny fontů se stejnou šířkou znaků (anglicky „monospaced“), kvůli možnosti přehlednějšího zarovnání informací pod sebe. Typografie je obecně poměrně rozsáhlý obor, ale mezi další nejdůležitější pojmy patří slitek [43] (ligatura), kdy jsou dvě písmena spojena do jednoho. Jeden z druhů úpravy mezer je označován pojmem kerning [42] (česky podřezávání) a cílem je zlepšení optického dojmu z textu, čehož se dosáhne zmenšením mezery mezi problematickými sousedícími písmeny (kerningový pár).

f | → fl

(a) Slitek

V A → VA

(b) Kerning

Obrázek 55: Typografické úpravy textu

Cílem práce ale nebylo navržení perfektního typografického algoritmu, proto byla implementována podpora pouze monospaced písma. Vybrány byly dva moderní fonty, Consolas a Lucida Console, oba jsou výchozí pro příkazovou řádku operačního systému Windows. Jsou navzájem velmi podobné, Lucida Console má o něco širší znaky. Z každého fontu bylo vygenerováno 10 velikostí jednoho řezu za použití programu The Dot Factory [38]. Podporovány jsou pouze znaky s ASCII hodnotou 0x20 – 0x7E.

Sázečí algoritmus funguje tak, že přes požadovaný text posunuje ukazatel (pointer) znak po znaku. Od každého znaku je odečten offset 0x20 (ASCII hodnota mezery) a rastrová reprezentace je dohledána ve vygenerované tabulce [38]. Znak je potom vykreslen na displej jako malý rastrový obrázek.

Ukázka použitých fontů Consolas a Lucida Console je níže na obrázcích obr. (56) a obr. (57)<sup>1</sup>.

```
0123456789;<=>?@!"#$%&'()*+,-./[\]^_`{|}~
The quick brown fox jumps over the lazy dog. 1234567890
The quick brown fox jumps over the lazy dog. 1234567890
The quick brown fox jumps over the lazy dog. 1234567890
```

Obrázek 56: Ukázka fontu Consolas

```
0123456789;<=>?@!"#$%&'()*+,-./[\]^_`{|}~
The quick brown fox jumps over the lazy dog. 1234567890
The quick brown fox jumps over the lazy dog. 1234567890
The quick brown fox jumps over the lazy dog. 1234567890
```

Obrázek 57: Ukázka fontu Lucida Console

<sup>1</sup> Anglická věta „The quick brown fox jumps over the lazy dog“ se často používá při testování fontů, protože obsahuje všechny znaky anglické abecedy.

## 6.2 BOARD SUPPORT PACKAGE DRIVER

V rámci diplomové práce bylo vytvořeno několik knihoven, které se souhrnně často označují jako „Board Support Package“ (BSP). Je zvykem výrobců dodávat je ke každé konkrétní desce pro zjednodušení obsluhy jednotlivých periférií.

Pro navrženou desku vznikly následující soubory:

BSP .....	<i>Kořenový adresář BSP.</i>
├ bsp_general.c (+.h) .....	<i>Definice LED, tlačítek, GPIO, COM, FTDI.</i>
├ bsp_ar1020.c (+.h) .....	<i>Ovladač pro dotykovou vrstvu.</i>
├ bsp_lcd.c (+.h) .....	<i>Grafické knihovny.</i>
├ bsp_sd.c (+.h) .....	<i>Ovladač pro SD kartu.</i>
├ bsp_sdram.c (+.h) .....	<i>Ovladač pro SDRAM.</i>
├ bsp_tpa2012.c (+.h) .....	<i>Ovladač pro zesilovač.</i>
└ bsp_wm8731.c (+.h) .....	<i>Ovladač pro audiokodek.</i>

Soubory BSP využívají API HAL knihoven 1.5 [22] (viz kap. 6, obr. 54). Soubor *bsp\_lcd* pro správnou činnost potřebuje vlastní vygenerované fonty (kap. 6.1). Dema byla navržena tak, aby se všechna odkazovala na jednu společnou složku s HAL, CMSIS a BSP knihovnami. Proto je nutné v cestách kompilátoru pro každý projekt zavádět následující společná umístění:

```

../../Inc
../../../../../Fonts
../../../../../BSP
../../../../../Drivers/STM32F4xx_HAL_Driver/Inc
../../../../../Drivers/STM32F4xx_HAL_Driver/Inc/Legacy
../../../../../Middlewares/Third_Party/FatFs/src/drivers
../../../../../Middlewares/Third_Party/FatFs/src
../../../../../Drivers/CMSIS/Include
../../../../../Drivers/CMSIS/Device/ST/STM32F4xx/Include

```

Dema byla testována ve vývojovém prostředí System Workbench for STM32 [32]. Pro správnou funkci je nutné nastavit přepínač „Debug Level“ se standardní hodnoty „Maximum -g3“ na „Default -g“. Jiné nastavení způsobovalo problémy s knihovnou FatFS [27]. Kompatibilita byla testována s přepínačem „Optimization Level“ na hodnotě „Optimize for size (-Os)“, ostatní varianty z časových důvodů odzkoušeny nebyly.

Do nově založeného projektu je třeba pro správnou funkci BSP knihoven a fontů přidat dvě virtuální složky. V příložených kódech jsou umístěny ve vývojovém prostředí v záložce „Project Explorer“.

/ .....	<i>Hlavní virtuální složka vývojového prostředí pro každé demo.</i>
├ Application .....	<i>Obsahuje mj. soubor main.c a definici interruptů.</i>
├ Debug .....	<i>Složka pro debugger, obsahuje mj. i výstupní .bin soubor.</i>
├ Drivers	
│ └ BSP .....	<i>Třeba vytvořit a importovat sem soubory BSP.</i>
│ └ CMSIS .....	<i>Obsahuje odkazy na CMSIS knihovny.</i>
│ └ Fonts .....	<i>Třeba vytvořit a importovat sem soubory s fonty.</i>
│ └ STM32F4xx_HAL_Driver .....	<i>Obsahuje odkazy na HAL knihovny.</i>
├ Middlewares .....	<i>Odkazy pro FatFS knihovnu.</i>
└ user .....	<i>Obsahuje soubor syscalls.c.</i>

## 6.3 DEMONSTRAČNÍ KÓDY PRO BSP

Jednotlivé soubory BSP knihovny (kap. 6.2) byly postupně odzkoušeny na několika jednoduchých demo kódech, které jsou popsány dále v kapitole. Byly vytvořeny snímky obrazovky („screenshoty“) použitého TFT LCD displeje ET0700G0DH6 [5], pořízené výpisem části SDRAM, která odpovídá frame bufferu konkrétní vrstvy.

Data jsou opatřena hlavičkou a uložena do formátu BMP (kap. 3.6). Vrstvy periferie LTDC (kap. 4.5) pracují s vnitřním kódováním barev ARGB888. To BMP formát bohužel nepodporuje, proto musel být odstraněn alfa kanál a uložení proběhlo v paletě True color (24 bit).

Ukázka kódu na této stránce demonstruje možný způsob, jak se dá udělat snímek obrazovky. Jedná se o zjednodušenou variantu – skutečná použitá implementace ošetřuje chyby, které mohou vzniknout při čtení a zápisu. Důležité je uvědomit si, že na displeji jsou barvy reprezentovány naopak, než jsou ve skutečnosti uloženy v souboru – například bílý odstín je na LCD vyjádřen maximálním jasem všech složek (0xFFFFFFFF), ale v souboru je odstín uložen jako součet nulových itenzit kanálů (0x000000).

Zvolený způsob vytvoření screenshotu je sice poměrně jednoduchý, ale není velmi výkonově efektivní. Výhodnější by byla konverze frame bufferu do kódování RGB888 přímo periferií LTDC (kap. 4.5) do jiné části SDRAM. Odtud by pak proběhl sekvenční zápis na SD kartu. Bohužel, tato varianta nebyla z důvodů časového presu realizována.

```
// vytvoreni souboru screenshotu
f_open(&MyFile, "SCREEN.BMP", FA_CREATE_ALWAYS | FA_WRITE);

// zapis hlavicky
f_write(&MyFile, (uint32_t*) mujheader, 54, (void *)&byteswritten);

// zapis pixel po pixelu
for(i=0; i<(800*480); i++){
    BSP_SDRAM_ReadData((0xC0000000 + (i*4)), &barva, 1);

    // zahodi se alpha a rozdeli se mezi kanaly
    barva&=0x00FFFFFF;
    barva_r = (uint8_t)((barva>>16)&0x000000FF);
    barva_g = (uint8_t)((barva>>8)&0x000000FF);
    barva_b = (uint8_t)(barva&0x000000FF);

    // kanaly se musi invertovat
    barva_r=0xFF-barva_r;
    barva_g=0xFF-barva_g;
    barva_b=0xFF-barva_b;

    // ulozeni kanalu pozpatku
    f_write(&MyFile, barva_b, (UINT)1, (void *)&byteswritten);
    f_write(&MyFile, barva_g, (UINT)1, (void *)&byteswritten);
    f_write(&MyFile, barva_r, (UINT)1, (void *)&byteswritten);
}

f_close(&MyFile);
```

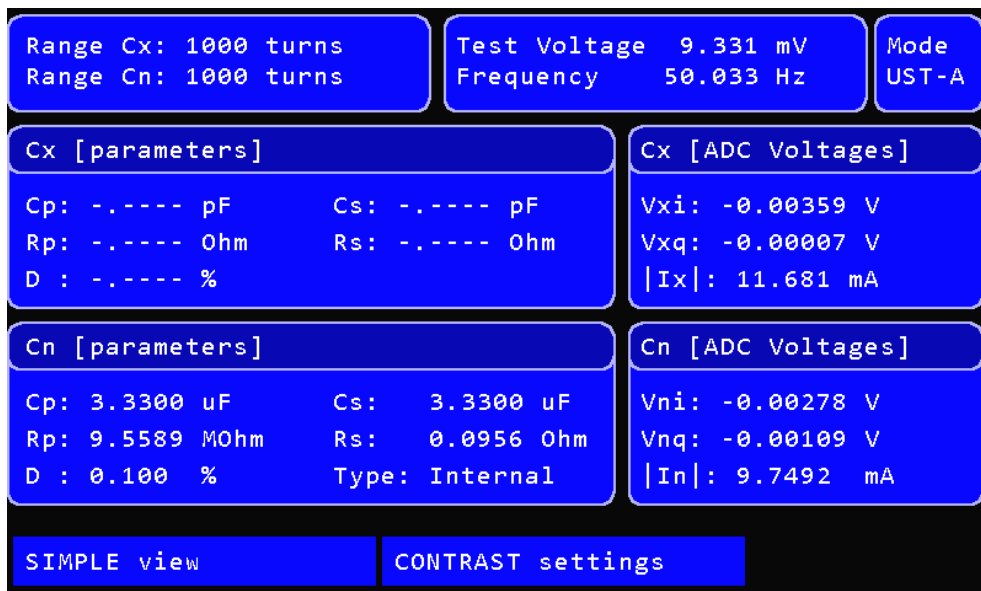


### 6.3.1 GUI DEMO

Pro demonstraci knihovny *bsp\_lcd.c* bylo vytvořeno demo, které má představovat ukázkou reálné aplikace. Demo simuluje uživatelské rozhraní, které je vyvíjeno na jednom z přístrojů na katedře měření ČVUT FEL na „konkureční“ platformě uLCD-70DT a komerčních knihovnách firmy 4D Systems [28] (kap. 2 obr. 3).

Obrázek obr. (58) ukazuje výsledek použití některých funkcí vlastní knihovny *bsp\_lcd.c*. Základ balíčku jsou hotové kódy firmy ST Microelectronics, odladěné pro vzniklou desku v rámci této práce a spojené do jednoho souboru. Při vývoji bylo čerpáno z BSP knihoven, které jsou určeny pro demonstraci desek STM324x9I\_EVAL, STM324xG\_EVAL a STM32446E\_EVAL a jsou součástí knihoven dodávaných k STM32CubeMX [21].

Soubor je rozšířen o podporu dalších vlastních primitiv, jako vykreslování oktantů kružnic pomocí Bresenhamova algoritmu (kap. 3.5), které slouží ke kreslení obdélníků se zaoblenými rohy. Knihovna využívá vlastní fonty (kap. 6.1), které jsou umístěné zvlášť ve složce „Fonts“.



Obrázek 58: Screenshot obrazovky dema GUI

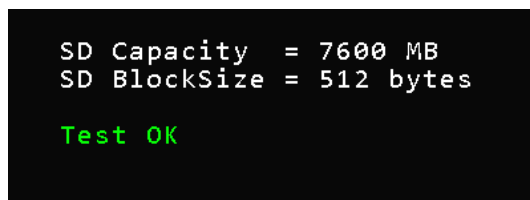
Použití knihovny je intuitivní. Následující kód demonstruje inicializaci a zapnutí displeje, který bude používat paměť (frame buffer) na adrese 0xC000 0000 (začátek SDRAM). Text „Hello world!“ bude vykreslený fontem Consolas velikosti 21 px bílou barvou na modrém pozadí. Přehled všech API funkcí je možné najít v hlavičkovém souboru *bsp\_lcd.h*.

```
BSP_LCD_Init();
BSP_LCD_DisplayOn();
BSP_LCD_LayerDefaultInit(0, 0xC0000000);
BSP_LCD_SelectLayer(0);
BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
BSP_LCD_SetFont(&Consolas21);
BSP_LCD_Clear(LCD_COLOR_BLUE);
BSP_LCD_DisplayStringAtLine(5, (uint8_t*)"Hello world!");
```

### 6.3.2 TEST PODPORY SD KARTY A KNIHOVNY FATFS

Další demo testuje implementaci FatFs [27] knihovny na použité paměťové kartě. Výstupem na displej je informace o kapacitě a velikost jednoho sektoru karty. Data jsou získána ze struktury *SDCardInfo*, která se automaticky naplní po inicializaci a zavolání funkce *BSP\_SD\_GetCardInfo()*;

```
sd_status = BSP_SD_Init();
if(sd_status == SD_OK){
    BSP_SD_GetCardInfo(&SDCardInfo);
    sprintf(buffer, "    SD Capacity = %d MB", SDCardInfo.CardCapacity/(1024*1024));
    BSP_LCD_DisplayStringAtLine(1, buffer);
    sprintf(buffer, "    SD BlockSize = %d bytes", SDCardInfo.CardBlockSize);
    BSP_LCD_DisplayStringAtLine(2, buffer);
}
```



Obrázek 59: Screenshot obrazovky testu microSD

Samotný test souborového systému FAT32 [24] probíhá tak, že je knihovně FatFs [27] nejprve nalinkován příslušný BSP ovladač *bsp\_sd.c* pro konkrétní desku. Reference na BSP (include ovladače) musí být i v souboru *ffconf.h*, který slouží pro konfiguraci FatFs a v příložených kódech k této práci je vždy uložen společně s demem ve složce *inc*. Dalším krokem je připojení a inicializace karty funkcí *f\_mount*.

Na kartě je následně vytvořen testovací soubor, který je naplněn daty (*f\_write*) a uzavřen (*f\_close*). V poslední části se soubor otevře, přečte, a pokud obsah souhlasí, demo je úspěšně ukončeno. Následující kód ukazuje zjednodušený postup použití FatFs. Demo ve skutečnosti výstup volání každého příkazu striktně kontroluje. Další funkce knihovny FatFs je možné dohledat na stránkách projektu [27].

```
// inicializace
FATFS_LinkDriver(&SD_Driver, SDPath) != 0);
f_mount(&SDFatFs, (TCHAR const*)SDPath, 0);

// vytvoreni a naplneni souboru
f_open(&MyFile, "TEST.TXT", FA_CREATE_ALWAYS | FA_WRITE);
f_write(&MyFile, wtext, sizeof(wtext), (void *)&byteswritten);
f_close(&MyFile);

// precteni sobouru
f_open(&MyFile, "TEST.TXT", FA_READ);
f_read(&MyFile, rtext, sizeof(rtext), (UINT*)&bytesread);
f_close(&MyFile);
```

### 6.3.3 TEST RYCHLOSTI SD KARTY

Výsledky měření v kapitole 4.8.4, obr. (24) byly získány z vytvořeného dema *sd\_benchmark*. Je použita FatFS [27] knihovna (kap. 6.3.2) a zjišťuje se rychlost čtení a zápisu různých bloků dat s různou velikostí clusterů na kartě (problematika organizace dat je popsána v kap. 4.8.3).

Metodika testování byla následující:

1. Inicializace karty a souborového systému.
2. Vytvoření souboru.
3. Vytvoření bloku dat s konkrétní velikostí (512 B, 1 kB, 2 kB, 4 kB, 8 kB, 16 kB, 32 kB, 64 kB).
4. Start měření zápisu (v přerušení systick, volané každou milisekundu).
5. Zapsání 100 bloků dat.
6. Stop měření zápisu.
7. Zavření a uložení souboru.
8. Otevření souboru.
9. Start měření čtení.
10. Čtení 100 bloků dat.
11. Stop měření čtení.
12. Zavření souboru.
13. Skok do bodu 3 a změna velikosti bloku.

Výsledky měření jsou graficky zobrazeny v kap. 4.8.4, obr. (24), ukázka výsledků dema na TFT LCD displeji, viz níže (obr. 60).

```

Block = 512 bytes
- write 51200 bytes in 300 ms, speed = 170 kB/s
- read 51200 bytes in 270 ms, speed = 189 kB/s
Block = 1024 bytes
- write 102400 bytes in 328 ms, speed = 312 kB/s
- read 102400 bytes in 305 ms, speed = 335 kB/s
Block = 2048 bytes
- write 204800 bytes in 375 ms, speed = 546 kB/s
- read 204800 bytes in 308 ms, speed = 664 kB/s
Block = 4096 bytes
- write 409600 bytes in 510 ms, speed = 803 kB/s
- read 409600 bytes in 317 ms, speed = 1292 kB/s
Block = 8192 bytes
- write 819200 bytes in 950 ms, speed = 862 kB/s
- read 819200 bytes in 364 ms, speed = 2250 kB/s
Block = 16384 bytes
- write 1638400 bytes in 716 ms, speed = 2288 kB/s
- read 1638400 bytes in 438 ms, speed = 3740 kB/s
Block = 32768 bytes
- write 3276800 bytes in 1642 ms, speed = 1995 kB/s
- read 3276800 bytes in 575 ms, speed = 5698 kB/s
Block = 65536 bytes
- write 6553600 bytes in 2225 ms, speed = 2945 kB/s
- read 6553600 bytes in 860 ms, speed = 7620 kB/s
*** BENCHMARK SUCCESSFUL ***

```

Obrázek 60: Screenshot obrazovky testu rychlosti microSD

Celý test je napsán jako jediný cyklus, velikosti bloků se počítají a přepínají automaticky. Výpočet bloku probíhá bitovým posunem a násobením základní velikostí sektoru 512 B.

```

for(i=0; i<8; i++){
    velikost_bloku = 512*(1<<j);
}

```

### 6.3.4 TEST RYCHLOSTI SDRAM + RAMDISK

Grafické jádro použitého mikrokontroléru sestává z akcelerátoru DMA2D (Chrom-Art Accelerator, kap. 4.4) a LCD-TFT periferie LTDC (kap. 4.5), které společně s Cortexovým jádrem přistupují do externí SDRAM (kap. 4.9.4) pomocí periferie FMC (Flexible Memory Controller, kap. 3.7 obr. 11).

V rámci dema *sdram\_benchmark* byly otestovány 3 možnosti, jak je možné mezi jádrem, akcelerátorem a SDRAM přenášet data.

První možnost je adresování dat přímo pomocí pointeru na adresu 0xC000 0000 (začátek banky SDRAM). Tento postup simuluje první možnost „polling mode“ (blokující mód), kdy je SDRAM postupně sekvenčně zapisována a čtena. Výhodnější varianta je použití DMA (Direct Memory Acces), kdy se data začnou kopírovat „na pozadí“ a Cortex jádro není vytíženo. Je tak dosaženo i vyšší rychlosti. Pro test byl použit balík 4 MB dat, rychlost byla měřena v přerušení systick.

Nejrychlejší varianta dema je využití grafického akcelerátoru pro zaplnění SDRAM obrazovými daty. K tomu byla použita funkce *BSP\_LCD\_Clear()*, která vyplní obrazovku jednolitou barvou, specifikovanou v parametru funkce. Bylo vyzorováno, že zapnutí či vypnutí displeje má vliv na rychlost, proto byly otestovány obě varianty.

Dosažený datový tok stačí i s rezervou na plynulé překreslování celé obrazovky s rozlišením 800 × 480 rychlostí 30 snímků za sekundu.

```
*** BENCHMARK SUCCESFULL ***
Polling mode:
- write in 175 ms, speed = 23 MB/s
- read in 305 ms, speed = 13 MB/s

DMA mode:
- write in 105 ms, speed = 39 MB/s
- read in 232 ms, speed = 18 MB/s

DMA2D mode (LCD is OFF):
- write in 941 ms, speed = 163 MB/s
- /read not measured/

DMA2D mode (LCD is ON):
- write in 1506 ms, speed = 101 MB/s
- /read not measured/

Benchmark consists of read/write 4MB data pack
and 100x LCD refresh with DMA2D with LCD ON/OFF
```

Obrázek 61: Screenshot obrazovky testu rychlosti SDRAM

Další demo zaměřené na SDRAM je *ramdisk*. Demonstruje možnost naformátování SDRAM na souborový systém FAT32, takže je možné v této rychlé paměti přímo pohodlně vytvářet a zapisovat soubory. To může být výhodné například při nutnosti měření s velmi vysokou vzorkovací frekvencí s mnoha kanály převodníku zároveň. Příkladem náročné aplikace mohou být třeba datalogery Magneti Marelli instalované do vozů Formule 1 [39].

Demo *ramdisk* je velmi podobné testu SD karty popsaném v kap. 6.3.3. Sestává z vytvoření souborového systému, založení a naplnění nového souboru daty a jejich zpětného přečtení.

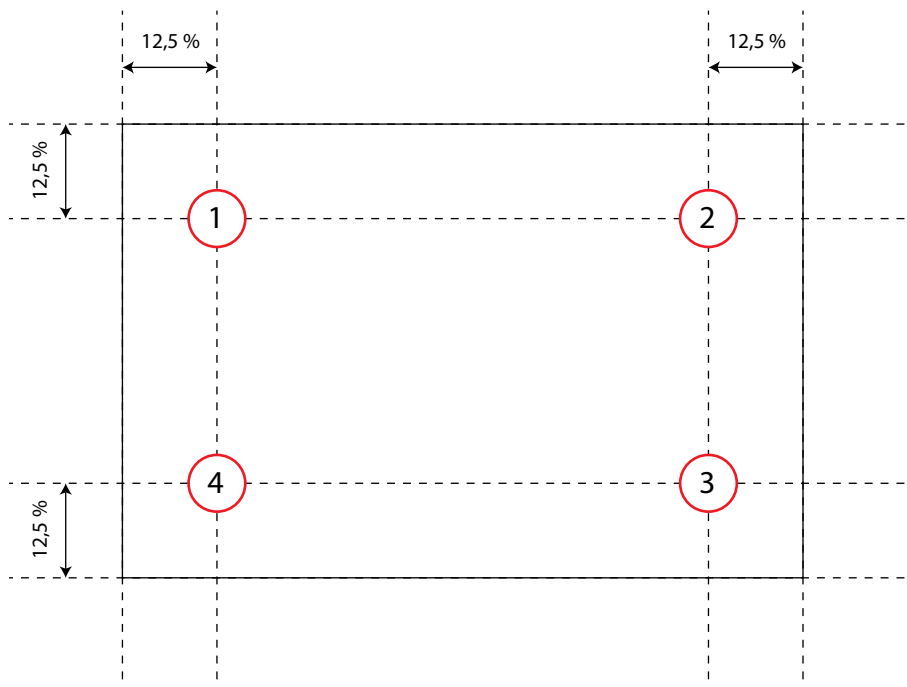
### 6.3.5 DEMO DOTYKOVÉ VRSTVY A KALIBRACE

Poslední demo *touch\_calibrate* ukazuje práci s dotykovou vrstvou a zaměřuje se taktéž na kalibraci. Vrstva funguje na rezistivním principu, kdy je dotyk prstu snímán pomocí elektrod okolo displeje a vniklé napětí vyhodnoceno A/D převodníky. Problematice se je více věnována kapitola kap. 3.4 (teorie) a kap. 4.7 (hardware). Každý panel ale může být trochu odlišný, proto je potřeba použítý řadič správně zkalibrovat.

Použitý řadič AR1020 [13] komunikuje po sběrnici I2C. Poloha dotyku je vyčítána „Touch Report“ protokolem, kalibrace je inicializována příkazem 0x14 „Command“ protokolu. Komunikace je rozebrána v kap. 4.7.1. Po inicializaci kalibrace řadič zapne interrupt pin a vyšle potvrzovací sekvenci <0x55><0x02><0x00><0x04>. Poté se čeká postupně na čtyři dotyky od uživatele, ve směru hodinových ručiček (na obr. 62 vyznačeno čísly). Dotyk je potvrzen sekvencí <0x55><0x02><0x00><0x14>.

Uživateli musí být v průběhu kalibrace postupně zobrazeny všechny značky a informace o kalibraci. K tomu jsou použity funkce knihovny *bsp\_lcd* na vykreslení teček v přesném místě dotyku a kružnice pro lepší orientaci, podobně jako na obrázku níže. Úspěšný průběh je signalizován zelenou obrazovkou. Výrobce řadiče stanovuje místa dotyku ve vzdálenosti 12,5 % od okraje. Při rozlišení použitého displeje ET0700G0DH6 [5] 800×480 je offset 100 pixelů horizontálně a 60 vertikálně z každé strany.

Po úspěšně dokončené kalibraci je v demu možné zkusit si volně kreslit po displeji.



Obrázek 62: Průběh kalibrace dotykové vrstvy

## 6.4 DOSTUPNÉ GRAFICKÉ KNIHOVNY PRO STM32F4XX

V dnešní době (2015) na trhu již existují komerčně a průmyslově používané grafické knihovny. Mezi nejlepší patří Touch GFX a STemWin, kterým je věnována tato kapitola. Bohužel, z nedostatku času již knihovny nebyly otestovány.

### 6.4.1 TOUCH GFX

Velmi zdařilá knihovna je Touch GFX [34] Dánské firmy Draupner Graphics A/S. Je kompletně napsaná v C++ a vyniká nízkými nároky na paměť. Podporuje rozlišení až  $1024 \times 600$  a plně využívá grafický akcelerátor řady STM32F4x7/9. Podporována je ale i dalšími jádry Cortex M0, M3 a M4.

Existují výukové desky, které vyrábí společnost NXP, ST (STM32F429 a STM324x9I-EVAL), Freescale, Renesas a Silicon Labs. Ukázky demo aplikací vytvořených s Touch GFX jsou na obr. (63) a obr. (64). Zdroj obrázků [45].



Obrázek 63: Demo počasí vytvořené s knihovnou TouchGFX



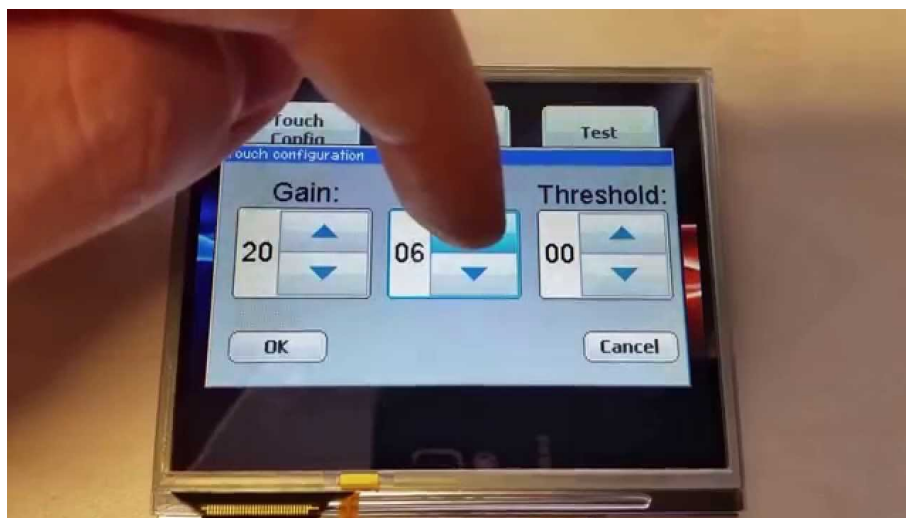
Obrázek 64: Demo termostatu vytvořené s knihovnou TouchGFX

Bohužel, licence není zdarma a proto není knihovna vhodná pro různé menší projekty. Cena při komerčním nasazení se pohybuje v rozmezí 3000 € – 15 000 € (80 – 400 tisíc Kč).

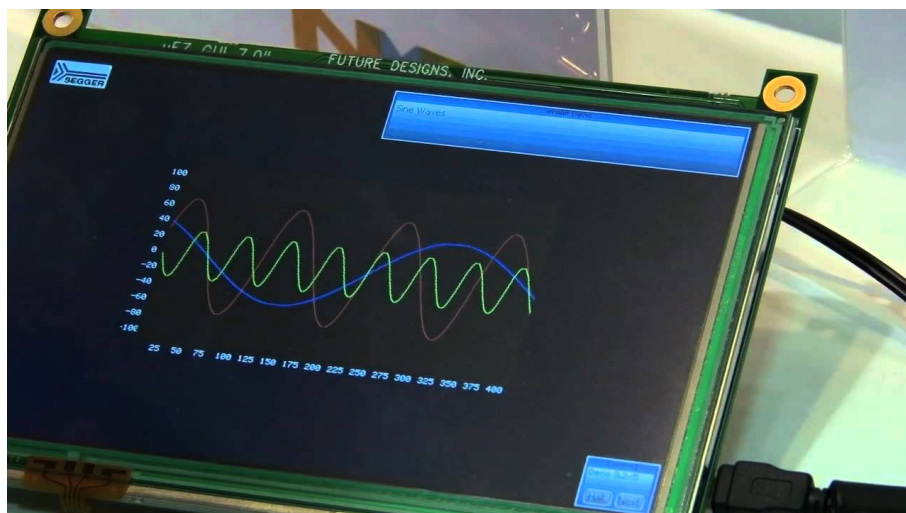
### 6.4.2 STEMWIN

Další kvalitní knihovna je STemWin [33], původně vytvořená firmou Segger pod názvem emWin. Vizuálně se snaží přiblížit operačnímu systému Windows, na který je majorita uživatelů zvyklá z počítačů. Knihovna je nyní portovaná pro mikrokontroléry ST Microelectronics a firma ji oficiálně podporuje a doporučuje. Pro vývoj je možné použít software emWin GUIBuilder, který je součástí balíčku STemWin.

Použití je vhodné pro malé projekty i školní účely. Licence je poskytována zdarma. Následující obrázky (obr. 65 a obr. 66) ukazují části aplikací, vytvořené s knihovnou STemWin. Zdroj obrázků [44].



Obrázek 65: Ukázka uživatelského rozhraní vytvořeného se STemWin



Obrázek 66: Graf vykreslený knihovnou STemWin





## 7 PLATFORMA PRO UCHYCENÍ DISPLEJE

Pro účely diplomové práce byla vyrobena platforma pro uchycení desky plošných spojů (viz přílohy obr. 71), programátoru a redukce (obr. 75) či CPLD redukce (obr. 73) ke zvolenému displeji. Podrobný výkres pro výrobu viz přílohy (obr. 69).

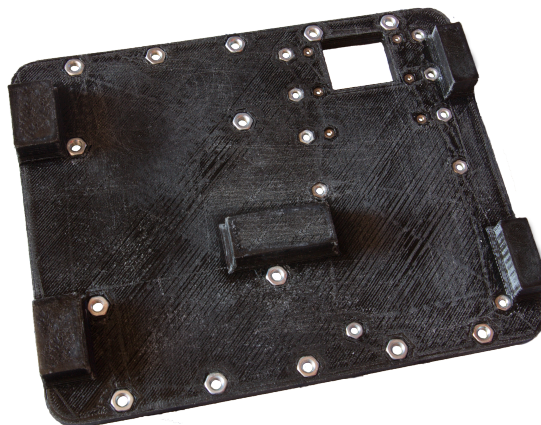
Platforma byla vyrobena technologií „3D tisk“ na tiskárně RepRap [31] (Replicating Rapid Prototyper). RepRap je 3D tiskárna, která tiskne nejen předměty, ale i vlastní součástky, takže je schopná se sama replikovat. Při tisku se vytváří trojrozměrné předměty postupným přidáváním materiálu (additive manufacturing), takže nevzniká žádný odpad.

Princip je takový, že se roztaví hmota (tenké vlákno), které se postupně nanáší na podložku, kde tuhne. Tisková hlava nakreslí vždy jednu vrstvu a posune se o kousek výš, kde začne kreslit další. Pro výrobu platformy bylo použito vlákno s průměrem 3 mm a tištěná vrstva měla výšku 300 mikrometrů. Po povrchovém opracování byly do šestiúhelníkových otvorů vlepeny montážní matky dvousložkovým epoxidem pro stabilnější připevnění plošných spojů nerezovými šrouby.

V současné době (2015) existují v podstatě 2 používané druhy materiálů, kterými se dá tisknout:

- ABS (Acrylonitrile Butadiene Styrene), běžně používaný termoplastický kopolymer s pracovní teplotou mezi 200 – 250° C. Je odolný vůči mechanickému poškození, nezávadný a při tavení má amorfní strukturu. ABS je možné rozpouštět acetonem a toluenem, čehož se využívá při vytvrzování a vyhlazování vytištěných dílů. Z tohoto materiálu je vyrobena i většina kostek LEGO.
- PLA (Polylactic acid) je bioplast vyrobený z biomasy. Na pohled a dotyk je stejný jako běžné syntetické plasty, oproti ABS je tvrdší (ale křehčí). Materiál nepředstavuje ekologickou zátěž, protože je vyroben z látek na základě škrobu. Ze škrobu je při vysokých teplotách izolována glukóza, ze které se kvašením získá kyselina mléčná a později kyselina polymléčná (právě polyactic acid, PLA). Teplota tavení je mezi 180 – 220° C. PLA není rozpustný v acetonu ani toluenu, takže je možné jej pouze obrušovat.

Platforma byla vyrobena z materiálu PLA.



Obrázek 67: Platforma pro uchycení desek k displeji



## 8 ZÁVĚR

Diplomová práce seznamuje čtenáře s problematikou konstrukce univerzálního TFT LCD řadiče, a to jak z pohledu teoretického, tak ze strany návrhu a vývoje hardware a software. Velký důraz byl kladen na všeobecné použití, proto byl v průběhu práce navržen a otestován způsob, jak připojit libovolný LCD panel. Zkonstruována a nakonfigurována byla aktivní, programovatelná redukce s CPLD a pasivní, jednoúčelová redukce pro konkrétní zvolený panel. Navržené metody se v průběhu práce ukázaly jako velmi efektivní, funkční a levné. Přínosem je taktéž použití univerzálního konektoru, do kterého se redukce zapojují, neboť je možné se současným zachováním hlavní řídicí desky přijít s dalším způsobem, jak připojit jiný panel (např. podpora VGA výstupu do počítače).

Práce na začátku v teoretické části shrnuje související problematiku, kterou bylo nutné před zahájením vývoje nastudovat. Jedná se především o funkci dotykové vrstvy, algoritmus pro vykreslování křivek a samotnou architekturu mikrokontroléru, daného zadáním. Druhá kapitola se věnuje převážně hardware a kompletnímu popisu schémat i funkčnosti. Důraz je kladen na grafický akcelerátor a jeho využití při vykreslování. Rozšíření paměti bylo dosaženo implementací paměťových karet a souborového systému FAT32. Příslušná kapitola obsahuje kompletní popis a test rychlosti, čímž částečně navazuje na bakalářskou práci autora. Následuje část věnovaná externí paměti SDRAM, která je použita k alokaci bufferu pro grafická primitiva. Dobré pochopení funkčnosti bylo nutné k optimalizaci řadiče paměti pro dosažení dobrých rychlostních výsledků, které jsou dále v kapitole shrnuty. Celé zařízení je doplněno několika komunikačními rozhraními a podporou audio výstupu.

Hardware doplňuje navazující kapitola, která je věnována popisu a otestování vytvořených knihoven. Vzniklo několik demo kódů, které jsou v textu podrobněji rozebrány, doplněny ukázkami zdrojových kódů a snímky obrazovky. Knihovny doplňuje sada fontů. Velké úsilí bylo věnováno studiu architektury ARM, protože autor měl v tomto ohledu před začátkem práce pouze minimální zkušenosti a nikdy vlastní desku s Cortex jádrem nenavrhol.

Všechny části původně stanoveného zadání byly splněny. Výsledkem diplomové práce je fyzicky vyrobené, oživené a otestované zařízení. Bohužel, průběh vývoje nebyl úplně hladký a vzhledem ke komplexnosti problematiky vzniklo několik drobných hardwarových chyb, které nebyly v průběhu práce z časových důvodů odstraněny, ale byly podrobně zdokumentovány. Jedná se především o nepozornost při návrhu pasivní redukce, která způsobovala chybnou interpretaci obrazové informace zahazováním nejvyšších dvou bitů, namísto nejnižších. Problém se podařilo obejít tak, že chybné polotóny v demo kódech nebyly použity a uživatelské rozhraní je tvořeno jenom základními barevnými kanály, které vadou netrpí. Druhá dokumentovaná chyba vznikla na desce CPLD v části připojení dotykové vrstvy, kdy byly prohozeny vodiče sběrnice I2C, což znemožnilo dotykové ovládání s připojenou CPLD redukcí. Odstranění obou chyb je poměrně triviální a nezamezuje funkčnosti samotné hlavní desky řadiče.

Do budoucna se počítá s vylepšováním a dalším vývojem celé platformy. Možné je nasazení například jako výukového prostředku ve škole, případně implementace do katedrou vyvíjených přístrojů. Bude třeba se ještě zaměřit na některé části, které byly implementovány nad rámec zadání a nebyly v průběhu diplomové práce z časových důvodů uspokojivě vyřešeny. Jmenovitě jde o otestování USB, lepší podporu audiokodeku a nasazení vhodného grafického prostředí, například STemWin.



## REFERENCE

- [1] STM32F43xxx advanced ARM®-based 32-bit MCUs (Reference manual) [online]. 18.3.2015  
[http://www.st.com/web/en/resource/technical/document/reference\\_manual/DM00031020.pdf](http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf)
- [2] YIU, Joseph. The definitive guide to ARM CORTEX-M3 and CORTEX-M4 processors. 3rd ed. Oxford: Elsevier, c2014, xxxv, 818 s. ISBN 978-0-12-408082-9.
- [3] Altera – MAX V Device Handbook [online]. 18.3.2015  
[https://www.altera.com/en\\_US/pdfs/literature/hb/max-v/max5\\_handbook.pdf](https://www.altera.com/en_US/pdfs/literature/hb/max-v/max5_handbook.pdf)
- [4] STM32F439xx ARM®Cortex-M4 32b MCU+FPU (Datasheet) [online]. 18.3.2015  
<http://www.st.com/web/en/resource/technical/document/datasheet/DM00077036.pdf>
- [5] Emerging Displays ET0700G0DH6 - TFT LCD datasheet [online]. 18.3.2015  
[http://www.glyn.de/Produkte/Displays/Aktive-Displays-\(TFT\)?q=G-ET0700G0DH6](http://www.glyn.de/Produkte/Displays/Aktive-Displays-(TFT)?q=G-ET0700G0DH6)
- [6] Micron SDR SDRAM, 64 Mb MT48LC4M16A2 datasheet [online]. 18.3.2015  
[https://www.micron.com/~media/documents/products/data-sheet/dram/64mb\\_x4x8x16\\_sdram.pdf](https://www.micron.com/~media/documents/products/data-sheet/dram/64mb_x4x8x16_sdram.pdf)
- [7] ISSI SDRAM, 512 Mb IS42S16320D datasheet [online]. 18.3.2015  
[http://www.issi.com/WW/pdf/42-45R-S\\_86400D-16320D-32160D.pdf](http://www.issi.com/WW/pdf/42-45R-S_86400D-16320D-32160D.pdf)
- [8] TPA2012 – Stereo Class-D Audio Power Amplifier datasheet [online]. 18.3.2015  
<http://www.ti.com/lit/ds/symlink/tpa2012d2.pdf>
- [9] MAX3232 – True RS-232 Transceiver (Datasheet) [online]. 18.3.2015  
<http://pdfserv.maximintegrated.com/en/ds/MAX3222-MAX3241.pdf>
- [10] TPD4E1B06 4-Channel Ultra Low Leakage ESD Protection Device [online]. 18.3.2015  
<http://www.ti.com/lit/ds/symlink/tpd4e1b06.pdf>
- [11] WM8731 - Portable Internet Audio Codec with Headphone Driver [online]. 18.3.2015  
<http://www.cs.columbia.edu/~sedwards/classes/2008/4840/Wolfson-WM8731-audio-CODEC.pdf>
- [12] LMZ21701 1A Nano Module 17V Max. Input [online]. 18.3.2015  
<http://www.ti.com/lit/ds/symlink/lmz21701.pdf>
- [13] AR1000 Series Resistive Touch Screen Controller [online]. 18.3.2015  
<http://ww1.microchip.com/downloads/en/DeviceDoc/41393A.pdf>
- [14] Active Displays (TFT) [online]. 18.3.2015  
[http://www.glyn.com/Products/Displays/Active-Displays-\(TFT\)](http://www.glyn.com/Products/Displays/Active-Displays-(TFT))
- [15] FTDI Drivers [online]. 18.3.2015  
<http://www.ftdichip.com/FTDrivers.htm>
- [16] Omron FPC/FFC Connector - XF Series [online]. 18.3.2015  
<http://www.omron.com/ecb/products/pdf/en-sdlj-069k.pdf>

- [17] Přehled nejčastějších barevných prostorů [online]. 18.3.2015  
[https://msdn.microsoft.com/en-us/library/windows/desktop/bb153349\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb153349(v=vs.85).aspx)
- [18] Image Quantization, Halftoning, and Dithering [online]. 18.3.2015  
<https://www.cs.princeton.edu/courses/archive/fall100/cs426/lectures/dither/dither.pdf>
- [19] Skript BMP2C, "Converts bitmap file to c/c++"[online]. 18.3.2015  
<http://sourceforge.net/projects/bmp2c/>
- [20] Bresenham's Circle Algorithm [online]. 18.3.2015  
[http://www.asksatyam.com/2011/01/bresenham-circle-algorithm\\_22.html](http://www.asksatyam.com/2011/01/bresenham-circle-algorithm_22.html)
- [21] STM32CubeMX for STM32 (User Manual) [online]. 18.3.2015  
[http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user\\_manual/DM00104712.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00104712.pdf)
- [22] Description of STM32F4xx HAL drivers [online]. 18.3.2015  
[http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/user\\_manual/DM00105879.pdf](http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/user_manual/DM00105879.pdf)
- [23] Oscillator design guide for STM8S, STM8A and STM32 microcontrollers [online]. 18.3.2015  
[http://www.st.com/web/en/resource/technical/document/application\\_note/CD00221665.pdf](http://www.st.com/web/en/resource/technical/document/application_note/CD00221665.pdf)
- [24] Microsoft Extensible Firmware Initiative FAT32 File System [online]. 18.3.2015  
[https://msdn.microsoft.com/en-us/library/windows/desktop/bb153349\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb153349(v=vs.85).aspx)
- [25] SD Physical Layer Simplified Specification Version 4.10 [online]. 18.3.2015  
[https://www.sdcard.org/downloads/pls/simplified\\_specs/part1\\_410.pdf](https://www.sdcard.org/downloads/pls/simplified_specs/part1_410.pdf)
- [26] BORÝSEK, Martin. Záznamové zařízení pro závodní elektromobil. Praha, 2013. Bakalářská práce. České vysoké učení technické, Fakulta elektrotechnická, Katedra měření.
- [27] FatFs - Generic FAT File System Module [online]. 18.3.2015  
[http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html)
- [28] 4D Systems - Intelligent Display Modules [online]. 18.3.2015  
<http://www.4dsystems.com.au/products>
- [29] GHI Electronics - System on Module [online]. 18.3.2015  
<https://www.ghielectronics.com/catalog/category/34>
- [30] Analog Devices ADV7123. Triple 10-Bit High Speed Video DAC [online]. 18.3.2015  
<http://www.analog.com/media/en/technical-documentation/data-sheets/ADV7123.pdf>
- [31] 3D tisk - RepRap tiskárna Silicon Hill [online]. 18.3.2015  
<http://3dtisk.sh.cvut.cz/cs>
- [32] SW4STM32 Software Development Environment for STM32 MCUs [online]. 18.3.2015  
<http://www.st.com/web/en/catalog/tools/FM147/CL1794/SC961/SS1533/PF261797>

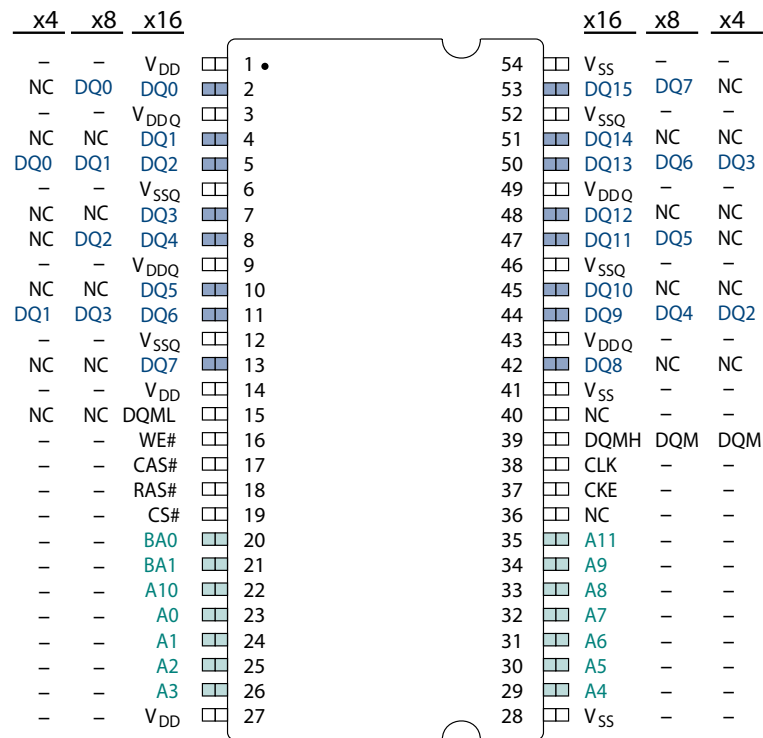
- [33] STemWin – Professional graphical stack library [online]. 18.3.2015  
<http://www.st.com/web/en/catalog/tools/FM147/CL1794/SC961/SS1743/LN1734/PF259225>
- [34] Grafická knihovna Touch GFX [online]. 18.3.2015  
<http://touchgfx.com/>
- [35] STM32 ST-LINK utility [online]. 18.3.2015  
<http://www.st.com/web/en/catalog/tools/PF258168>
- [36] How it works: 4-Wire Analog-Resistive Touch Screens [online]. 18.3.2015  
<https://www.sparkfun.com/datasheets/LCD/HOW%20DOES%20IT%20WORK.pdf>
- [37] dafont.com, kategorie fontů s fixní šířkou [online]. 18.3.2015  
<http://www.dafont.com/theme.php?cat=503>
- [38] The Dot Factory: An LCD Font and Image Generator [online]. 18.3.2015  
<http://www.eran.io/the-dot-factory-an-lcd-font-and-image-generator/>
- [39] Magneti Marelli datalogger HFD [online]. 18.3.2015  
[http://www.magnetimarelli.com/business\\_areas/motorsport/data-acquisition,-display-&-lap-trigger/data-logger#tab--3](http://www.magnetimarelli.com/business_areas/motorsport/data-acquisition,-display-&-lap-trigger/data-logger#tab--3)
- [40] Porovnání displaye IPS vs TN vs AMOLED [online]. 18.3.2015  
<http://www.smart-mobil.eu/clanky/6-porovnani-displaye-ips-vs-tft-tn/>
- [41] Samsung Galaxy S III's HD Super AMOLED display [online]. 18.3.2015  
<http://www.engadget.com/2012/05/03/galaxy-s-iii-microscope-hd-super-amoled/>
- [42] Design Fundamentals: Kerning and Tracking [online]. 18.3.2015  
<http://www.graphics.com/article-old/design-fundamentals-kerning-and-tracking>
- [43] Typografický slovníček [online]. 18.3.2015  
<http://www.typo.cz/database/pravidla-a-nazvoslovi/typograficky-slovnicek/>
- [44] 3.5" TFT + Segger emWin + STM32F429 [online]. 18.3.2015  
<https://www.youtube.com/watch?v=DGmumTezHAs>
- [45] Renesas newsletter - Embedded World 2015 Special Edition [online]. 18.3.2015  
[http://www2.renesas.eu/ap\\_newsletter/085\\_February-2015/index.html](http://www2.renesas.eu/ap_newsletter/085_February-2015/index.html)
- [46] Altia Design [online]. 18.3.2015  
<http://www.altia.com/release11.1/>
- [47] Display Modules With an FPGA and Embedded Processor [online]. 18.3.2015  
[https://www.altera.com/en\\_US/pdfs/literature/wp/wp-01074-creating-low-cost-intelli.pdf](https://www.altera.com/en_US/pdfs/literature/wp/wp-01074-creating-low-cost-intelli.pdf)
- [48] 750 PPI od Samsungu překonáno. Sharp má 5,5" 4k displej s 806 PPI [online]. 18.3.2015  
<http://diit.cz/clanek/sharp-4k-806-ppi-displej>





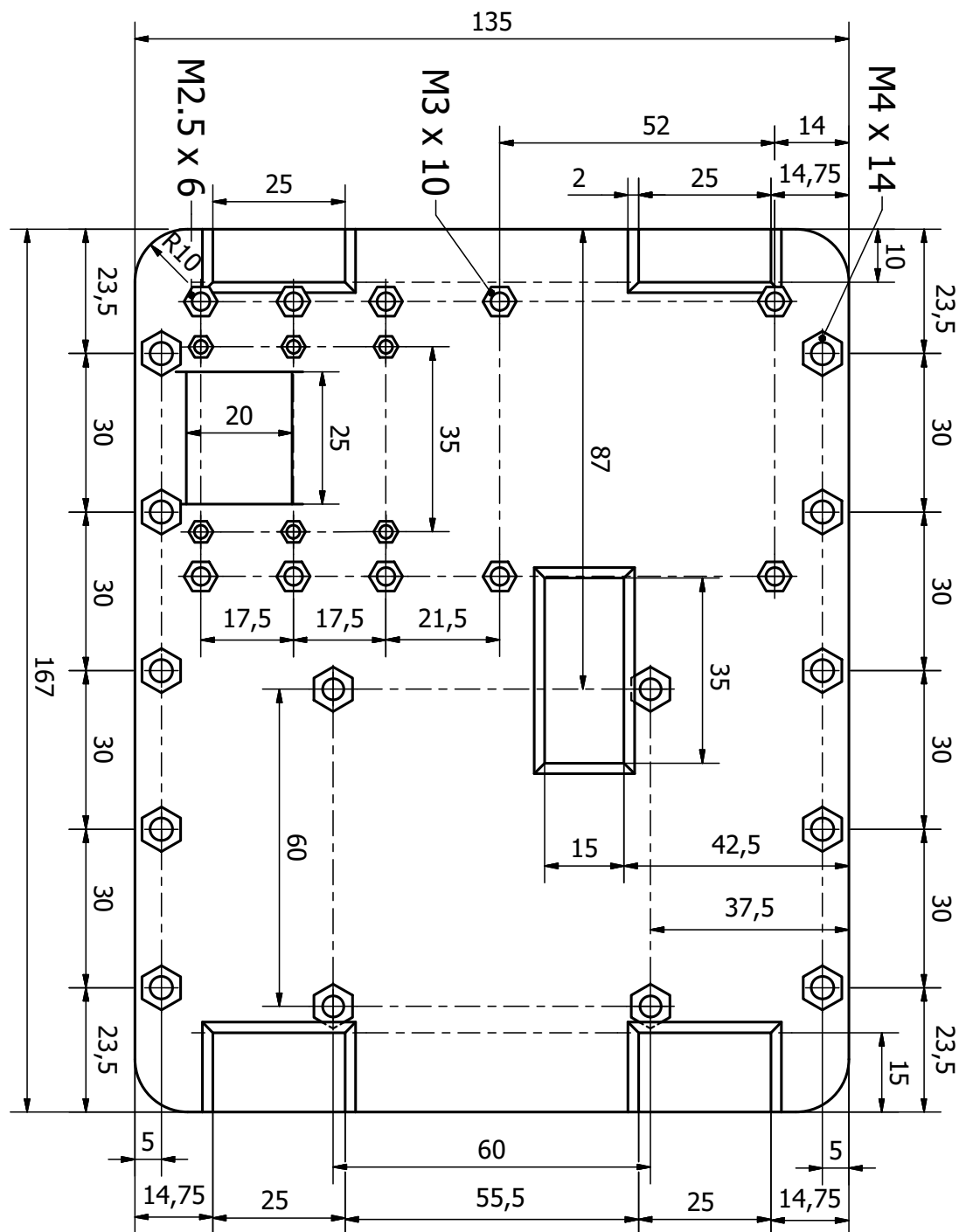
## 9 PŘÍLOHY

### 9.1 ROZLOŽENÍ PINŮ PAMĚTI SDRAM TSOPII (54 PIN)



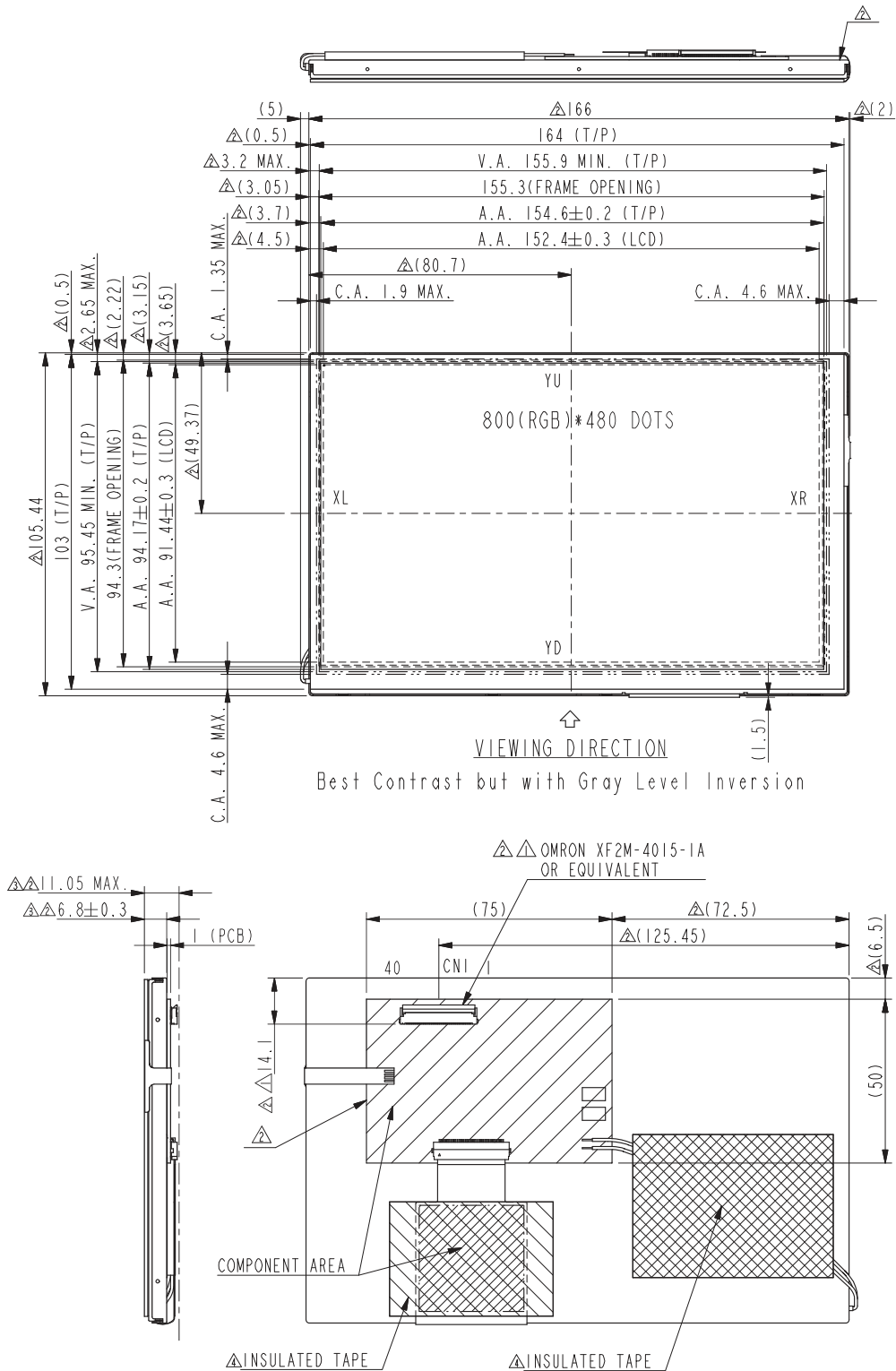
Obrázek 68: Rozložení pinů paměti SDRAM TSOPII 54 pin

## 9.2 PLATFORMA PRO UCHYCENÍ PCB K DISPLEJI



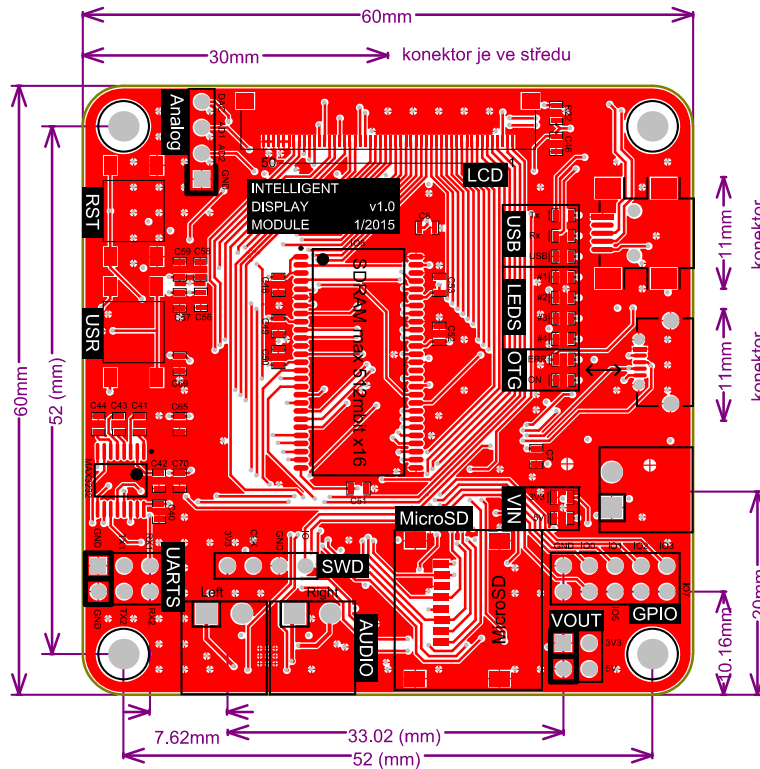
Obrázek 69: Výkres platformy pro uchycení PCB k displeji

### 9.3 NÁKRES POUŽITÉHO DISPLEJE ET0700G0DH6

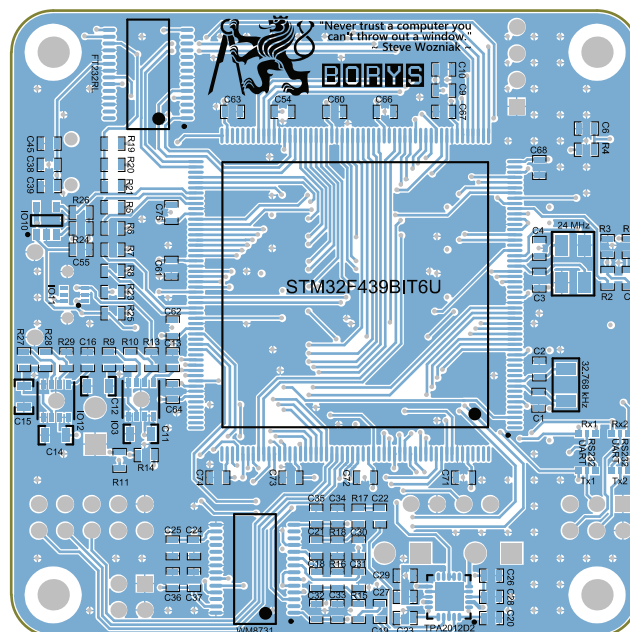


Obrázek 70: Rozměry displeje ET0700G0DH6

## 9.4 PLOŠNÝ SPOJ HLAVNÍ DESKY

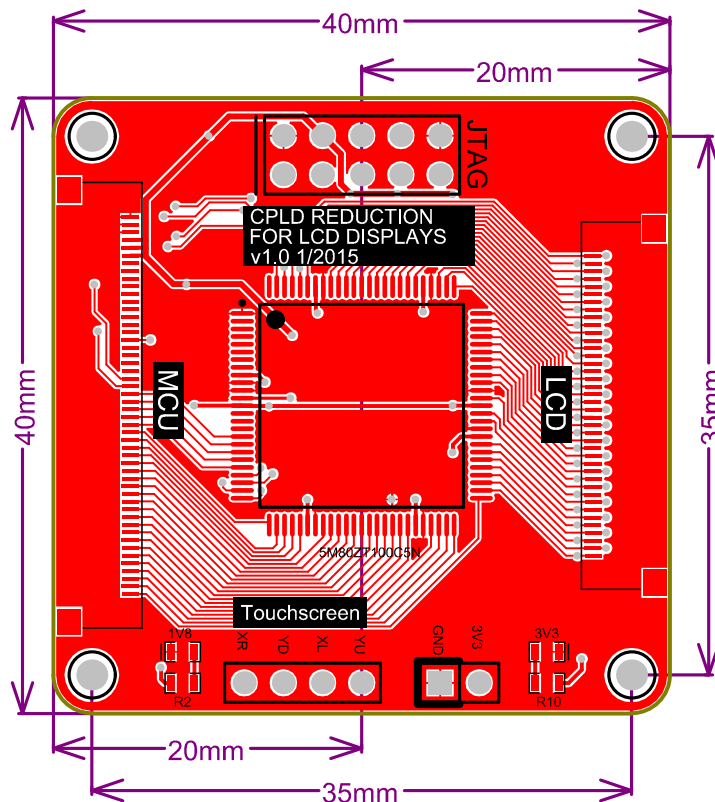


Obrázek 71: Hlavní deska - horní vrstva

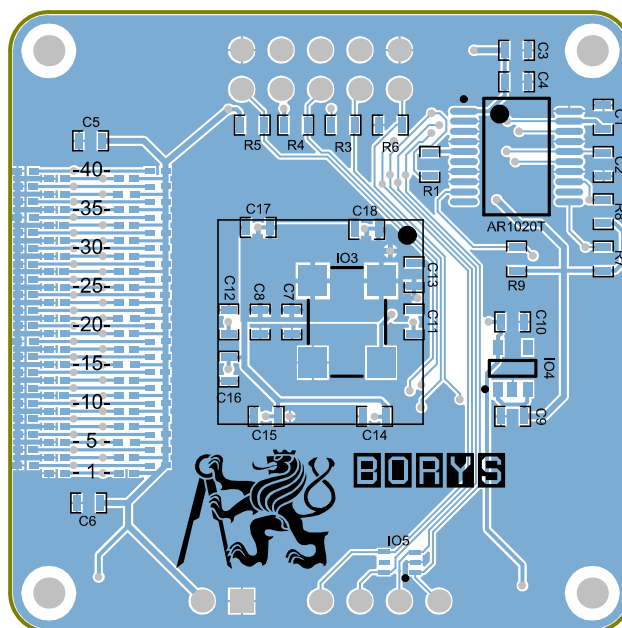


Obrázek 72: Hlavní deska - dolní vrstva

## 9.5 PLOŠNÝ SPOJ CPLD

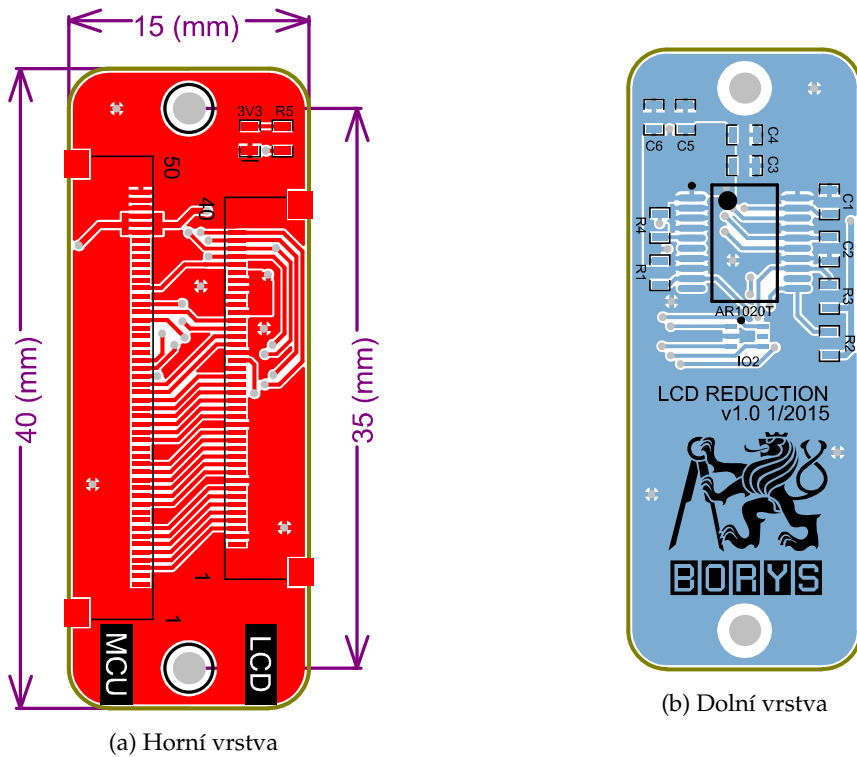


Obrázek 73: CPLD - horní vrstva



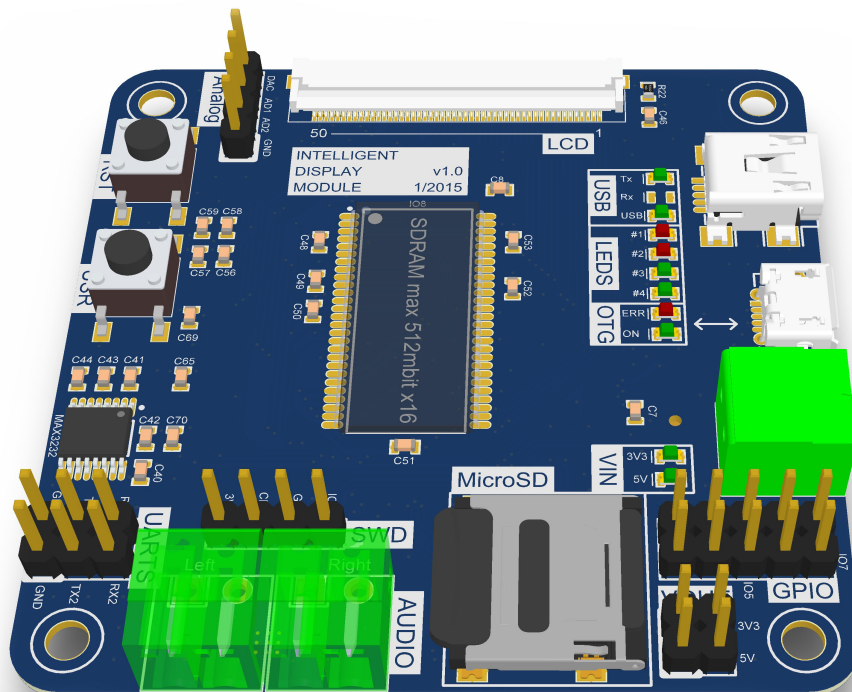
Obrázek 74: CPLD - dolní vrstva

## 9.6 PLOŠNÝ SPOJ REDUKCE

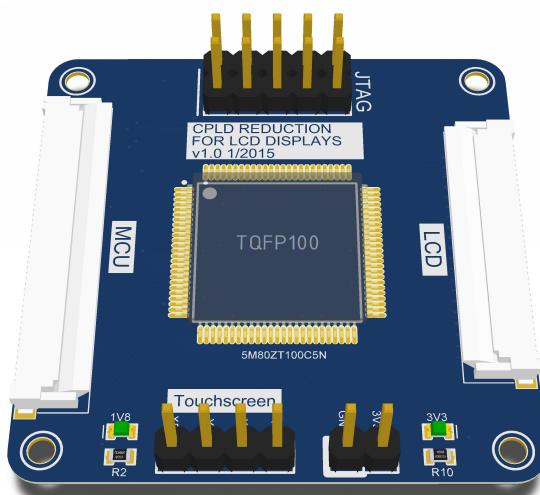


Obrázek 75: Redukce pro TFT LCD ET0700G0DH6

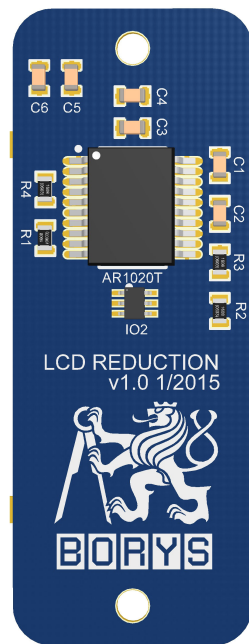
## 9.7 RENDER HOTOVÉHO ZAŘÍZENÍ



Obrázek 76: Render hlavní desky řadiče (horní strana)



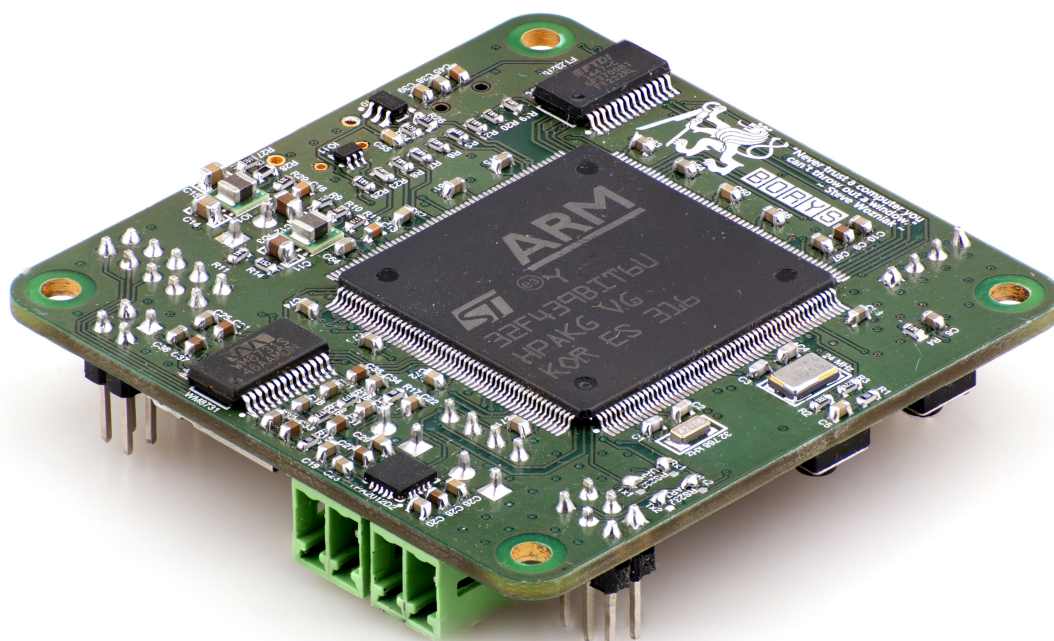
Obrázek 77: Render redukce s CPLD (horní strana)



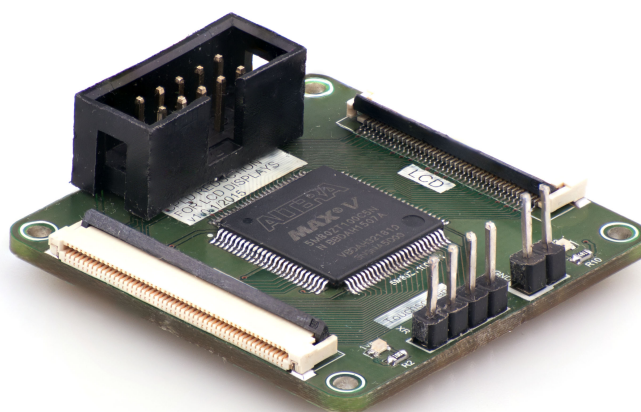
Obrázek 78: Render pasivní redukce (spodní strana)



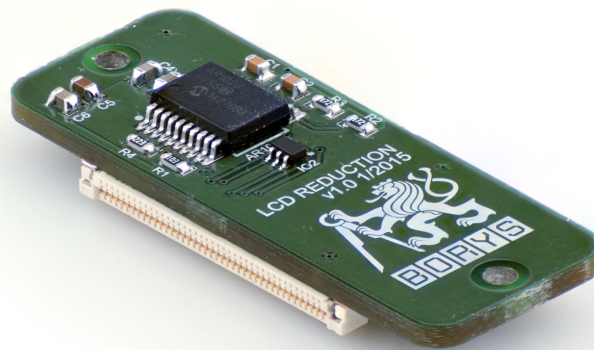
## 9.8 FOTOGRAFIE HOTOVÉHO ZAŘÍZENÍ



Obrázek 79: Hlavní deska řadiče (spodní strana)



Obrázek 80: Redukce s CPLD (horní strana)



Obrázek 81: Pasivní redukce (spodní strana)

## 10 OBSAH PŘILOŽENÉHO CD

/	
├── Kod	..... Hlavní složka se zdrojovým kódem.
│   ├── BSP	..... Board Support Package (BSP) napsaný v rámci této práce.
│   ├── Drivers	..... Složka s ovladači od STM Microelectronics.
│   │   ├── CMSIS	..... Cortex Microcontroller Software Interface Standard
│   │   │   ├── Device	..... Nízkoúrovňové definice pro mikrokontrolér (řada STM32f4xx).
│   │   │   ├── DSP_Lib	..... Knihovny pro digital signal processing.
│   │   │   ├── Include	..... Definice registrů a maker pro preprocesor.
│   │   │   └── RTOS	..... Základní API pro RTOS.
│   │   └── STM32F4xx_HAL_Driver	..... Hardware Abstraction Layer (HAL) v1.5.
│   ├── Fonts	..... Připravené fonty pro demo kódy.
│   ├── Middlewares	..... STemWin, Audio, USB, FatFS, FreeRTOS.
│   ├── Projects	..... Projekty, vzniklé v rámci této práce.
│   │   ├── demonstration	..... Zobrazení fontů, obrázku.
│   │   ├── gui_demo	..... Graphic User Interface (GUI) demo, kap. 6.3.1.
│   │   ├── paint_calibrate	..... kalibrace a test dotykové vrstvy, kap. 6.3.5.
│   │   ├── ramdisk	..... Namapování SDRAM jako RAMdisk, kap. 6.3.4.
│   │   ├── sd_benchmark	..... Test rychlosti paměťové karty, kap. 6.3.3.
│   │   ├── sd_test	..... Test kompatibility paměťové karty, kap. 6.3.2.
│   │   └── sdram_benchmark	..... Test rychlosti SDRAM, kap. 6.3.4.
├── diplomova_prace_borysek.pdf	..... Soubor s diplomovou prací
├── 3D modely	..... Složka s 3D modely.
└── Schemata	..... Složka se schématy.