



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

DIGITÁLNÍ BEZDRÁTOVÝ PŘENOS ZVUKOVÉHO SIGNÁLU

DIGITAL AUDIO WIRELESS TRANSMISSION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jeroným Juráň

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Zoltán Szabó, Ph.D.

BRNO 2017



Diplomová práce

magisterský navazující studijní obor Elektronika a sdělovací technika
Ústav radioelektroniky

Student: Bc. Jeroným Juráň

ID: 154753

Ročník: 2

Akademický rok: 2016/17

NÁZEV TÉMATU:

Digitální bezdrátový přenos zvukového signálu

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte pár vývojových desek umožňujících digitální přenos audio signálu ve standardu WISA. Vybavte analogovými i digitálními vstupy a výstupy. Ovládání realizujte pomocí barevného displeje s dotykovou vrstvou o úhlopříčce nejméně tři palce s rozlišením QVGA nebo vyšším. Využijte cenově dostupný 32 bitový mikrokontrolér architektury ARM. Vytvořte ukázkovou aplikaci demonstrující možnosti vyvinutých zařízení, konfiguraci spojení a ovládání většiny dostupných parametrů spojení. Požadované je přehledné grafické rozhraní. Použijte font včetně diakritiky, demonstруйте možnost vykreslování bitmap a umožněte konfiguraci barev uživatelského rozhraní.

DOPORUČENÁ LITERATURA:

[1] YIU, J. The definitive guide to ARM(R) Cortex(R)-M3 and Cortex-M4 processors. Third edition. ISBN 0124080820.

[2] Wireless Speaker and Audio (WiSATM) Association: White Paper. Homeautomationinfo [online]. [cit. 2016-0-18]. Dostupné z: http://www.homeautomationinfo.com/Drupal/files/WiSA_whitepaper_0809.pdf.

Termín zadání: 6.2.2017

Termín odevzdání: 16.5.2017

Vedoucí práce: Ing. Zoltán Szabó, Ph.D.

Konzultant:

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

UPOZORNĚNÍ:

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

ABSTRAKT

Dokument popisuje realizaci vývojových desek určených k vývoji aplikací pro bezdrátový přenos audio signálu pomocí WiSA kompatibilních modulů. Popisuje možnosti standardu WiSA, navržená schémata vývojových desek vysílače a přijímače a externích desek AD a DA převodníků. Dále shrnuje návrh desek plošných spojů zmíněných obvodů. Výsledky jsou na závěr demonstrovány pomocí fotodokumentace. Další část se zabývá návrhem knihovny obsluhující grafický řadič FTDI FT800 (a kompatibilních variant) a knihovny pro ukládání dat do FLASH paměti. Dále je popsána typická struktura aplikace využívající tento řadič včetně dotykového ovládání. Práce na závěr demonstruje dosažené výsledky a možnosti grafického řadiče na ukázkové aplikaci.

KLÍČOVÁ SLOVA

FT800, bezdrátový přenos, audio, WiSA, dotykové ovládání, FTDI, audio DAC, audio ADC

ABSTRACT

This document describes the design of evaluation boards for wireless transmission of audio signal. The boards use WiSA compliant modules. It describes schematics of transmitter board, receiver board and boards of both AD and DA converter. Secondly the document describes the designed PCBs and demonstrates the results by photodocumentation. It also includes description of C library for using FLASH memory to store data and the C driver for FT800 chip and the typical application using this chip including touch control. Document also describes application used for demonstration of the results.

KEYWORDS

FT800, wireless transmission, audio, WiSa, touch control, FTDI, audio DAC, audio ADC

JURÁŇ, J. *Digitální bezdrátový přenos zvukového signálu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2017. 16 s., 9 s. příloh. Diplomová práce. Vedoucí práce: Ing. Jan Juráň, Ing. Martin Friedl, Ph.D, Ing. Zoltán Szabó, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Digitální bezdrátový přenos zvukového signálu jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval Ing. Janu Juráňovi za vedení mé práce a praktické rady během vývoje. Dále pak Ing. Martinu Friedlovi, Ph.D a Ing. Zoltánu Szabó, Ph.D. za formální vedení a rady týkající se organizace a tvorby textové části práce.

OBSAH

Seznam obrázků	viii
Úvod	1
1 Teoretický úvod	2
1.1 Bezdrátový přenos audio signálu	2
1.1.1 WiSA	2
1.1.2 Komunikační moduly WiSA	4
1.2 Grafický řadič	4
1.2.1 Grafický řadič FT800.....	4
1.3 Další obvody	5
1.3.1 Mikrokontrolér	5
1.3.2 AD/DA převodníky.....	5
2 Návrh hardwaru	6
2.1 Návrh zapojení	6
2.1.1 Vysílač	6
2.1.2 Přijímač	8
2.1.3 AD/DA Převodníky	8
2.2 Návrh DPS	10
3 Návrh Softwaru	16
3.1 Ovladač pro FT800	16
3.1.1 Závislost na hardwaru	17
3.1.2 Struktura ovladače	17
3.1.3 Hlavní modul	18
3.1.4 Konfigurace ovladače	20
3.1.5 Struktura aplikací	21
3.1.6 Příklady GUI.....	23
3.2 Knihovna pro využití paměti FLASH.....	25
3.2.1 Princip činnosti	25
3.2.2 Použití knihovny	26
3.2.3 Shrnutí možností knihovny	26
3.3 Ukázková aplikace	27

3.3.1	Menu Wireless transmission	28
3.3.2	Menu FT800 demonstration.....	29
Závěr		30
Literatura		31
Seznam symbolů, veličin a zkratk		32
A Schémata		33
A.1	Schéma vývojové desky vysílače	33
A.2	Schéma vývojové desky přijímače	34
A.3	Schéma AD převodníku.....	35
A.4	Schéma DA převodníku.....	36
B Zobrazení vodivých vrstev		37
B.1	TX top	37
B.2	TX bot	38
B.3	RX top.....	39
B.4	RX bot.....	40
B.5	ADC/DAC	41

SEZNAM OBRÁZKŮ

Obr. 1 Konfigurace systému pomocí aplikace [1]	3
Obr. 2 Automatická konfigurace systému [1].....	3
Obr. 3 Konektor pro připojení externí ADC/DAC DPS	6
Obr. 4 Blokové schéma vývojové desky vysílače	7
Obr. 5 Blokové schéma vývojové desky přijímače	8
Obr. 6 Blokové schéma desky AD převodníku	9
Obr. 7 Blokové schéma desky DA převodníku	9
Obr. 8 3D zobrazení návrhu DPS vysílače shora.....	11
Obr. 9 3D zobrazení návrhu DPS vysílače zespoda	11
Obr. 10 3D zobrazení návrhu DPS AD a DA převodníku.....	12
Obr. 11 Hotová vývojová deska vysílače – pohled shora.....	12
Obr. 12 Hotová vývojová deska vysílače - pohled zespoda	13
Obr. 13 Hotová vývojová deska přijímače - pohled shora	13
Obr. 14 Hotová vývojová deska přijímače - pohled zespoda	14
Obr. 15 Hotové DPS AD a DA převodníku.....	14
Obr. 16 Hotová sestava vysílače.....	15
Obr. 17 Hotová sestava přijímače.....	15
Obr. 18 Příklad použití ovladače (widget hodiny, použití uživatelského fontu, bitmapy)	23
Obr. 19 Příklad použití ovladače (widgety tlačítek a klávesnice)	24
Obr. 20 Příklad použití ovladače (widget posuvníku)	24
Obr. 21 Vstupní obrazovka ukázkové aplikace	27
Obr. 22 Obrazovka pro nastavení hlasitosti	28
Obr. 23 Obrazovka pro nastavení barev	29

ÚVOD

Cílem práce bylo navrhnout a realizovat vývojový kit pro vývoj aplikací využívajících bezdrátový přenos audio signálu ve standardu WiSA. Kit se skládá z vysílače a přijímače, přičemž obě strany jsou vybaveny dotykovým displejem řízeným pomocí grafického řadiče FT800. Aby bylo možné kit využívat i pro testování AD a DA převodníku, jsou obě zařízení vybavena konektorem pro připojení externího převodníku. Práce obsahuje návrh dvoukanálového ADC modulu a čtyř kanálového DAC modulu.

Dalším cílem bylo seznámit se a získat zkušenosti s mikrokontroléry s jádrem architektury ARM. Pro tento účel byly zvoleny cenově dostupně mikrokontroléry STMicroelectronics.

První část práce stručně shrnuje možnosti audio přenosu v pásmu ISM. Dále pak popisuje zvolené integrované obvody. Druhá část pak popisuje navržený hardware včetně fotodokumentace. Následuje popis knihovny obsluhující grafický řadič FT800 a popis vyvinuté struktury kódu umožňující budoucí snadný a rychlý vývoj komerčních aplikací. Další část softwaru je tvořena knihovnou pro ukládání dat do FLASH paměti mikrokontroléru. Na závěr jsou výsledky práce demonstrovány pomocí ukázkové aplikace.

1 TEORETICKÝ ÚVOD

Následující kapitola stručně popisuje standard WiSA a použité moduly. Dále shrnuje možnosti vybraných použitých obvodů.

1.1 Bezdrátový přenos audio signálu

Pro bezdrátový přenos audio signálu se využívá více různých přístupů podle konkrétních požadavků.

Typickým zástupcem přenosu audio signálu na delší vzdálenosti je rozhlasové vysílání. Pro analogové vysílání se využívá AM (*amplitude modulation*) a FM (*frequency modulation*), pro digitální pak DAB (*Digital audio broadcasting*). V případě dnes už příliš nevyužívaném vysílání AM se využívá nosné okolo 1MHz. Přenos rozhlasového vysílání s modulací FM je pak přenášen v pásmu okolo 100MHz. Další variantou plošného vysílání je DAB. Jde o digitální vysílání v pásmech okolo 200MHz případně 1,5GHz. Tento standard je specifický především možností přenosu zvuku v CD kvalitě. [1]

Pro audio přenos lze využít sítě GSM. Tato služba je nicméně zpoplatněna a především neumožňuje přenos zvuku ve vysoké kvalitě.

Pro přenos na menší vzdálenosti se pak využívá zejména pásma ISM 2,4GHz nebo 5GHz. Velmi rozšířený je standard Bluetooth nebo WiFi. Nově se rozšiřujícím standardem je WiSA, který je určen právě pro přenos zvukového signálu ve vysoké kvalitě.

1.1.1 WiSA

Standard WiSa (*Wireless Speaker and Audio*) je určený pro přenos audio signálu především pro domácí sestavy. Hlavním přínosem je vysoká kvalita přenosu, robustní řešení, jednoduchá instalace, možnost rozšíření systému a kompatibilita mezi různými výrobci.

WiSA umožňuje přenos 24-bitového signálu bez komprese vzorkovací frekvencí 32 KHz, 44.1 KHz, 48 KHz a 96 KHz. Přenášeno je až 8 kanálů s pevným zpožděním 5 ms a zpožděním mezi jednotlivými kanály do 160 ns, což zaručuje synchronizaci s obrazem při sledování videa.

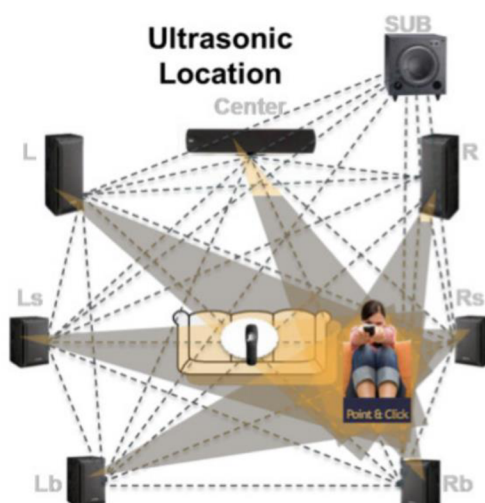
Robustnost přenosu je dána několika faktory. Je využíváno poměrně volné pásmo UNII pásmo 5.2 až 5.8 GHz s dynamickým přidělováním kanálů, díky čemuž si systém vybírá vhodný kanál. Navíc je k datům přidána redundance pro opravu případných chyb a pro další zvýšení kvality jsou případná chybící data inteligentně nahrazována tak, aby byly chyby nepostřehnutelné. Všechny WiSA produkty jsou testovány na ploše 10 x 10 metrů včetně překážek.

Drátové systémy je složité v místnosti rozmístit, vodiče jsou často vedeny podél zdí

nebo pod koberci. V případě dostatku financí pak skrz zdi nebo pod podlahou. Díky tomu je pak komplikované provádět jakékoliv změny v konfiguraci systému nebo stěhování nábytku v místnosti. Naproti tomu bezdrátové řešení stačí rozestavět libovolně po místnosti a připojit k napájení. V případě levnějších zařízení se typicky konfigurace provádí pomocí aplikace, díky které se jednotlivým reproduktorům nastaví kanál a relativní pozice v místnosti (viz Obr. 1). Systém pak sám upraví hlasitosti a fáze pro nejlepší posluchačský zážitek. Dražší systémy pak obsahují nadzvukové měniče („pingery“), díky kterým se na principu echolokace systém sám automaticky nastaví (viz Obr. 2).



Obr. 1 Konfigurace systému pomocí aplikace [1]



Obr. 2 Automatická konfigurace systému [1]

Kompatibilita se standardem WiSA zaručuje kompatibilitu zařízení mezi výrobci. Díky tomu je možné využívat vysílač a jednotlivé reproduktory od libovolných výrobců a tedy i sestavu si během času měnit a vylepšovat po částech.

Pro realizaci vývojových desek byly zvoleny moduly od firmy Summit. Konkrétně vysílací modul SWM908XD Tx a přijímací modul WS997XD Rx (viz kapitola 1.1.2). [2]

1.1.2 Komunikační moduly WiSA

Přenos je realizován pomocí páru SWM908XD Tx a WS997XD Rx. Oba moduly jsou WiSA certifikovány a tedy splňují všechna kritéria popsaná v kapitole 1.1.1. Moduly jsou interně řízeny pomocí ASIC obvodů a vyžadují dvojí napájení (1.2V a 3.3V).

Vysílací modul je řízen pomocí I²C, přičemž je podporován zpětný kanál pro přenos dat od přijímače k vysílači. Modul má integrovanou dvojici antén, přičemž umožňuje připojení externích antén. Modul je do hlavní desky připojen pomocí 40 pinového konektoru s roztečí 2.54 mm. Rozměry celého modulu jsou přibližně 9 cm x 3.5 cm.

Přijímač má integrované 4 antény. Připojen je pomocí 24 pinového konektoru. Rozměry jsou 7 cm x 4.5 cm.

Vzhledem k dohodě o mlčenlivosti vztahující se k těmto modulům práce nepopisuje žádné neverejné informace týkající se těchto modulů.

[3][4][5][6]

1.2 Grafický řadič

Tato podkapitola shrnuje nejzákladnější možnosti zvoleného grafického řadiče.

1.2.1 Grafický řadič FT800

Zařízení využívají grafického řadiče FT800. Tento obvod je specifický tím, že umožňuje tvorbu pokročilých GUI (grafických uživatelských rozhraní) s minimálními nároky na řídicí mikrokontrolér. Obvod je připojován pomocí SPI (*serial peripheral interface*) s rychlostí až 30MHz nebo pomocí i2c (*Inter-Integrated Circuit*) do rychlosti 3.4MHz. Varianta FT800 umožňuje připojení displejů o rozlišení až 512x512 pixelů, přičemž každá barva je reprezentována 6-ti vodiči. Jde tedy o 18-bitovou hloubku (262144 barev). Obvod dále podporuje připojení rezistivní dotykové vrstvy. Uživatelé pak poskytují nejen souřadnice stlačeného bodu, ale také informaci o objektu na daných souřadnicích. Řadič dále umožňuje generování PWM mono audio signálu a je tedy vhodný i pro složitá GUI.

Malé náročnosti pro řídicí mikrokontrolér je dosaženo mimo jiné pomocí widgetů, což jsou složitější prvky jako například tlačítka, posuvníky, hodiny, stavové ukazatele a mnohé další. Uživatelská aplikace může každému vykreslenému prvku nebo objektu přiřadit tag, který slouží jako identifikátor objektu. Následným sledováním tagu může aplikace zjišťovat, zda došlo k dotyku objektu a libovolným způsobem reagovat.

Pro vykreslení obrazovky se vytváří DL (*Display-list*). Pro jeho tvorbu je vymezená paměť velikosti 8192B, do které může mikrokontrolér zapisovat příkazy a následně je nechat vykreslit pomocí dedikovaného registru. Do této paměti může ale zapisovat také grafický koprocessor, který vytvoří widgety. Proto není vhodné do této paměti zapisovat a využívat raději FIFO koprocessoru k vykreslování jednoduchých prvků i widgetů. Tento blok paměti má velikost 4096B a pro jeho obsluhu má řadič dva registry sloužící jako zapisovací a čtecí ukazatel. Zapisovací ukazatel musí po zapsání

příkazů inkrementovat aplikace. Naopak čtecí ukazatel obsluhuje grafický koprocessor. Jsou-li si oba ukazatele rovné, znamená to, že všechny příkazy už byly zpracovány a zapsány do DL. FIFO je realizováno jako kruhový zásobník. [7]

1.3 Další obvody

1.3.1 Mikrokontrolér

Pro projekt byl zvolen cenově dostupný mikrokontrolér STM32F030CCT6, což je 32-bitový mikrokontrolér od firmy ST Microelectronics. Mikrokontrolér využívá jádra ARM Cortex M0 s taktem do 48MHz. Specifikem tohoto typu jsou relativně velké paměti, což je pro vývojové desky využívající grafický displej důležitým aspektem. Paměť RAM má 32kB, paměť typu FLASH je 256kB. Mikrokontrolér ve 48 pinovém pouzdře je vybaven následujícími komunikačními rozhraními: 2 x I2C, 2 x SPI a 6 x UART. Dalšími užitečnými obvody jsou RTC pětikanálové DMA a 11 časovačů. Programování a debugování je zajištěno pomocí rozhraní SWD. [8]

1.3.2 AD/DA převodníky

Převodníky jsou umístěny na samostatných deskách plošných spojů (DPS) a k vývojovým deskám se tak připojují jako externí obvody pomocí definovaného konektoru. Tento přístup umožňuje využití vývojových desek k testování různých převodníků.

AD převod je realizován pomocí AK5386, což je dvoukanálový 24-bitový audio převodník pracující na delta-sigma principu. Vzorkovací frekvence je v rozsahu 8 KHz až 216 KHz, přičemž jsou podporovány režimy *MSB justified* a i^2s . Výrobce udává poměr S/N 96 dB a rozsah DR 110 dB. Obvod má integrovaný antialiasingový filtr. Výstupní formát je konfigurovatelný pomocí vyvedených pinů. [8]

DA převod využívá 24-bitový delta-sigma převodník AK4430. Vzorkovací kmitočet je v rozsahu 8 KHz až 196 KHz. Podporované formáty vstupních dat jsou stejně jako u AD převodníku *MSB justified* a i^2s . Udávané zkreslení je THD+N -91dB a DR je 104dB. Převodník má integrovaný vstupní antialiasingový digitální FIR filtr. Výstupní filtr převodníku je externí dolní RC propust prvního řádu. [9]

Aby bylo možné snadno testovat a demonstrovat funkci, používá DA modul sluchátkový zesilovač MAX4410. Pro tento dvoukanálový zesilovač je specifické nízké napájecí napětí 1.8V až 3.6V. Obvod je realizován jako dvojitý operační zesilovač. Výstup každého kanálu má parametry 80 mW a 16 Ω . [11]

2 NÁVRH HARDWARU

Následující kapitola popisuje navržený hardware. Nejdříve popisuje zapojení jednotlivých DPS. Následně popisuje rozložení obvodů na DPS. Rozložení je demonstrováno pomocí fotodokumentace.

2.1 Návrh zapojení

2.1.1 Vysílač

Obvod vývojové desky vysílače je řízen mikrokontrolérem STM32F030CC programovaném pomocí rozhraní SWD. Mikrokontrolér pomocí SPI řídí grafický řadič FT800, ke kterému je dále připojen grafický RGB displej s dotykovou vrstvou. Pro buzení podsvícení je použit kontrolér TSP61041 zapojený jako zdroj proudu. Schéma zapojení FT800 a přilehlých obvodů bylo převzato od zadávající firmy.

Mikrokontrolér dále přes i^2c komunikuje s WiSA kompatibilním vysílacím modulem Summit SWM908XD TX. Tyto obvody jsou dále propojeny pomocí rozhraní UART a několika dalších pomocných signálů. Signály rozhraní i^2s , které mohou být při některém módu výstupy, jsou pro eliminaci odrazů vybaveny přizpůsobovacími rezistory. Metoda rezistorů umístěných o zdroje signálu byla využita proto, že tažení cesty požadované impedance (cca 60 až 80 Ω) by bylo nepraktické vzhledem k jejich značné šířce.

Pro připojení externího obvodu ADC je zařízení opatřeno definovaným konektorem (viz Obr 3). i^2c je připojeno k mikrokontroléru. i^2s je s čtyřmi datovými vodiči pro přenos až osmi kanály připojeno k vysílacímu modulu.

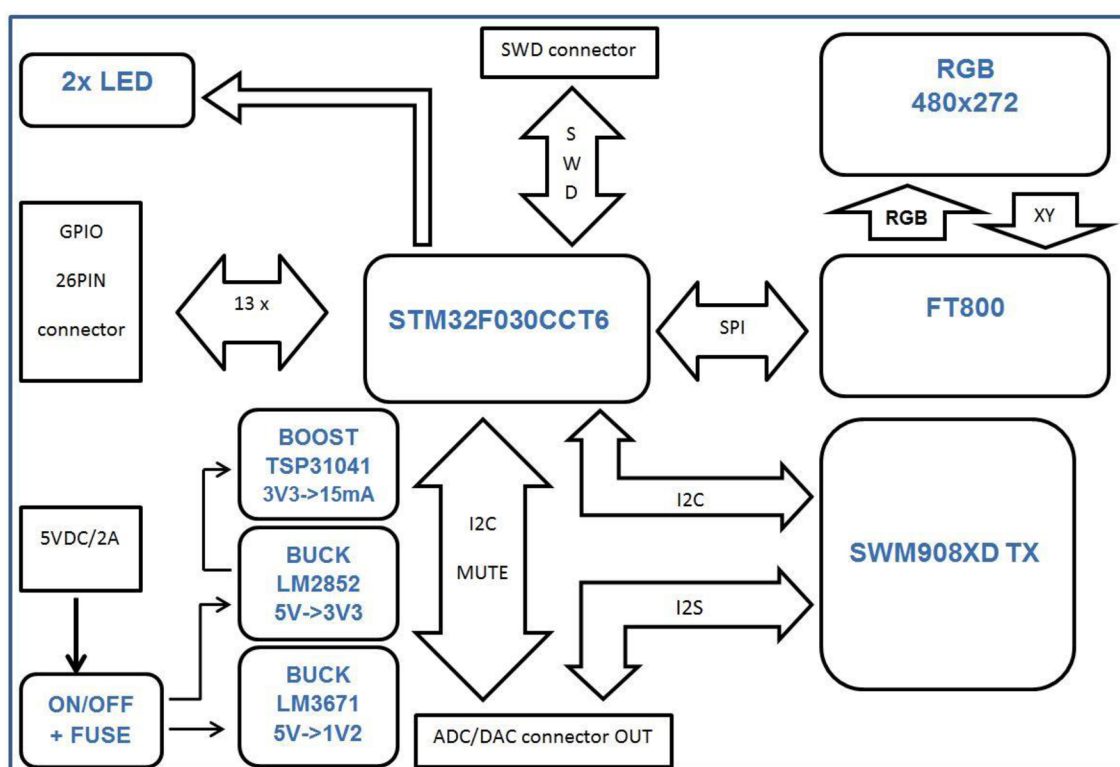
1	MUTE	GND	2
3	SDA	GND	4
5	SCL	GND	6
7	I2S_SCLK	GND	8
9	I2S_LRCK	GND	10
11	I2S_MCLK	GND	12
13	I2S_D3	GND	14
15	I2S_D2	GND	16
17	I2S_D1	GND	18
19	I2S_D0	GND	20
21	3V3	AGND	22
23	5V	AGND	24

Obr. 3 Konektor pro připojení externí ADC/DAC DPS

Dále jsou osazeny dvě LED pro libovolné použití. Pro dosažení maximální univerzálnosti zařízení bylo vyvedeno 13 GPIO pinů s ochranou proti připojení nevhodné úrovně nebo zkratu.

Zařízení je napájeno z externího síťového stejnosměrného adaptéru 5V s požadovaným proudem 2A. Z tohoto napětí se pomocí snižujícího měniče s kontrolérem LM2852X-3.3, což je pevný kontrolér s integrovaným spínačem, generuje napětí 3.3V. Toto napětí je využíváno jako hlavní napájecí napětí pro všechny další obvody na desce a tudíž je zdroj navržen na výstupní proud 1.5A. Dále se 5V využívá pro generování 1.2V pro vysílací modul. K tomuto účelu slouží snižující pevný měnič LM3671MT-1.2, který je rovněž vybaven integrovaným spínačem.

Blokové schéma vývojové desky vysílače je zobrazeno na následujícím obrázku (Obr 4). Kompletní schéma je umístěno v příloze (A.1).



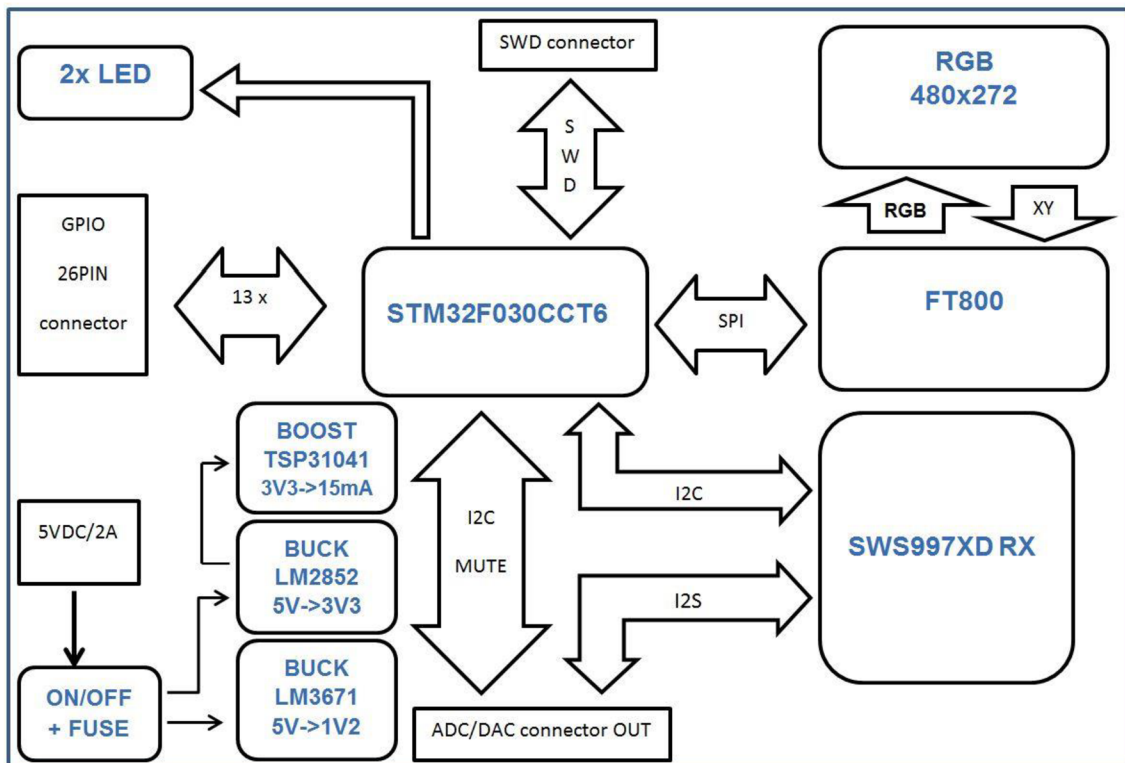
Obr. 4 Blokové schéma vývojové desky vysílače

2.1.2 Přijímač

Obvod vývojové desky přijímače vychází z obvodu vývojové desky vysílače jen s několika změnami.

Přijímač je osazen WiSA kompatibilním modulem Summit SWS997XD RX. Přijímač se tak liší od vysílače také připojením jiných pomocných signálů. Dále je zařízení osazeno tlačítkem pro párování s vysílačem.

Blokové schéma vývojové desky přijímače je zobrazeno na následujícím obrázku (Obr 5). Kompletní schéma je umístěno v příloze (A.2).



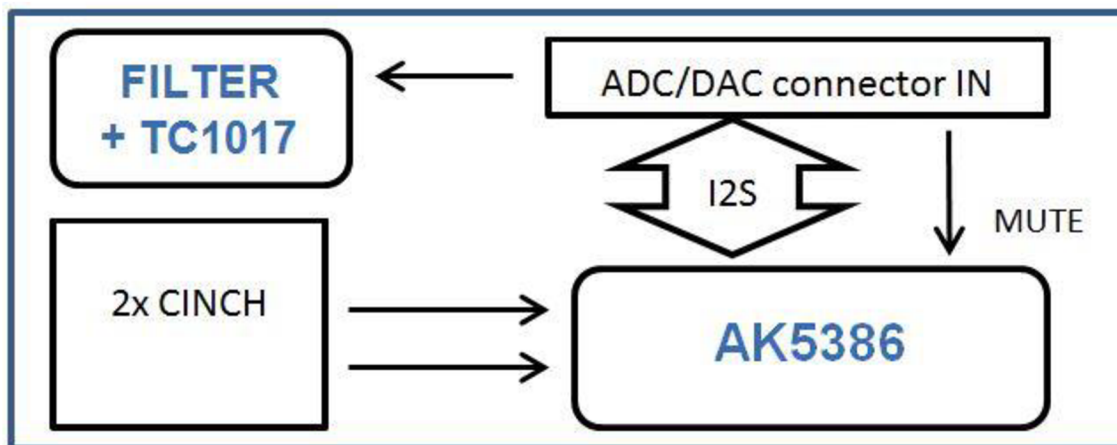
Obr. 5 Blokové schéma vývojové desky přijímače

2.1.3 AD/DA Převodníky

Převodníky jsou umístěny na externí DPS aby bylo možné vývojové desky použít také pro testování s různými převodníky. Pro připojení slouží definovaný konektor (Obr. 3). Verze konektoru na deskách převodníků má stejnou schematickou značku, ale odlišný footprint tak, aby bylo možné obě DPS správně propojit. Pro minimalizaci rušení, které by mohlo pronikat do výsledného signálu, není využíváno napájení 3.3V z konektoru, ale získává se z filtrovaného napětí 5V pomocí lineárního stabilizátoru TC1017-3.3VCT.

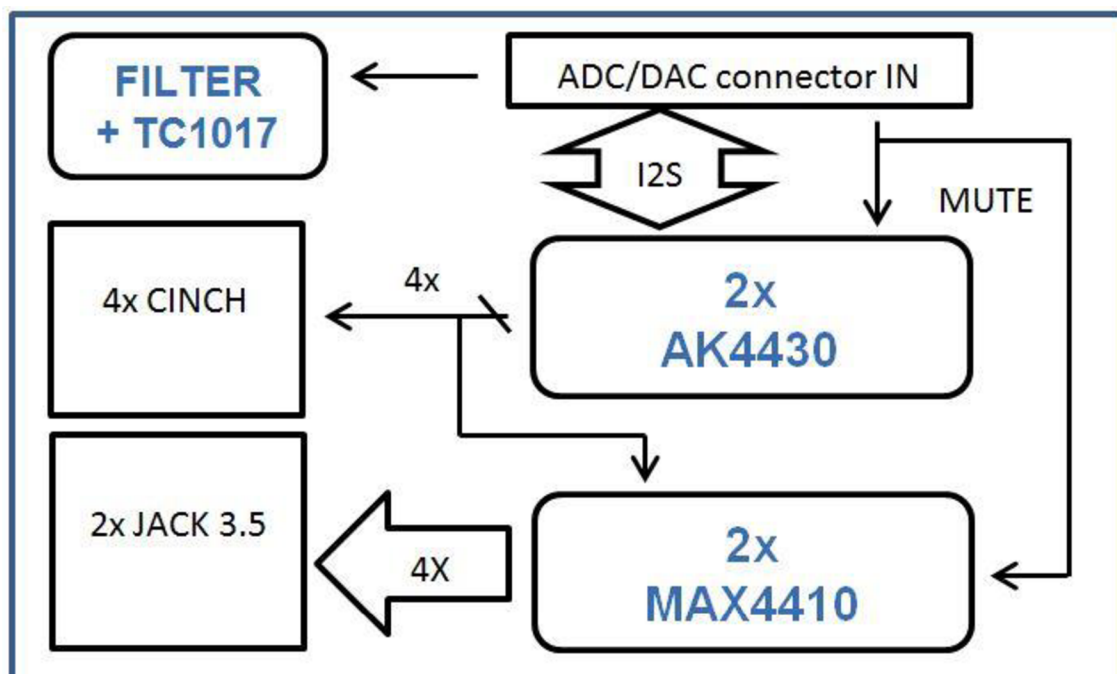
Deska AD převodníku obsahuje dva konektory typu CINCH. Za konektory jsou umístěny audio vazební kondenzátory 47uF, což je hodnota, která spolu s vstupním odporem vstupů převodníku tvoří horní propust s mezní frekvencí v řádu jednotek Hz. Díky tomu není vazební kapacitou ovlivněn vstupující zvukový signál. Obvod využívá dvoukanalový AD převodník AK5386. Blokové schéma analogově digitálního

převodníku je zobrazeno na následujícím obrázku (Obr 6). Kompletní schéma je umístěno v příloze (A.3).



Obr. 6 Blokové schéma desky AD převodníku

Deska DA využívá dvoukanalových převodníků AK4430. Díky tomu mohou být výstupem až čtyři kanály a obvod je tedy opatřen čtyřmi konektory typu CINCH. Pro usnadnění vývoje budoucích aplikací a snazší demonstrací možností vývojových desek, obsahuje deska také dva dvoukanalové sluchátkové zesilovače MAX4410 a dva výstupní konektory typu JACK 3.5. Blokové schéma digitálně analogového převodníku je zobrazeno na následujícím obrázku (Obr. 7). Kompletní schéma je umístěno v příloze (A.4).



Obr. 7 Blokové schéma desky DA převodníku

2.2 Návrh DPS

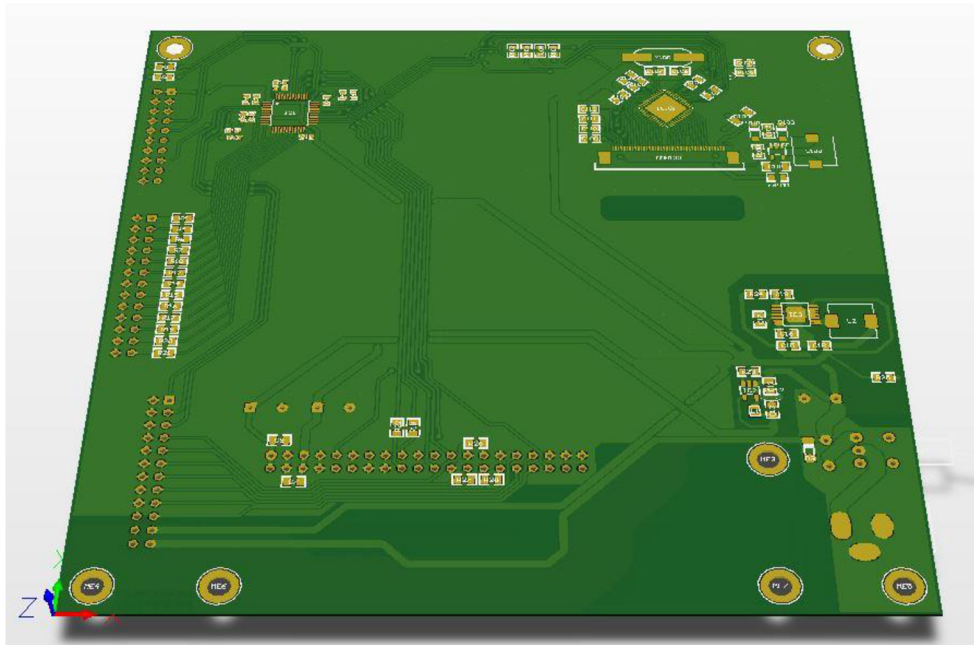
Desky plošných spojů byly navrženy podle zásad pro EMC (elektro magnetickou kompatibilitu). Plošné spoje jsou dvouvrstvé, přičemž spodní strana slouží především jako zemnicí plocha. Tato vrstva byla využívána pro tažení jiných spojů jen v případě nutnosti a její přerušení bylo realizováno tak, aby mělo minimální dopad na EMC. Každý funkční blok DPS je napájen samostatnou cestou vycházející z centrálního bodu za vstupním kondenzátorem. Veškeré funkční bloky byly sestaveny v souladu s doporučeními výrobcem daného integrovaného obvodu. Spínané zdroje jsou realizovány tak, aby bylo v maximální míře omezeno šíření rušení do okolí. Jelikož zařízení obsahuje audio části umístěné na externích DPS, je ke konektorům vedena oddělená analogová zem. Nejmenší šířka cesty byla 10 mil. Preferovaná šířka pak byla 16 mil, která byla použita také pro spoje i^2s . Jelikož tyto spoje by měly mít nižší charakteristickou impedanci, bylo přizpůsobení zajištěno přidáním rezistorů ke zdrojům těchto signálů. Návrh preferuje pouzdra 0603 a 0805.

Desky jsou koncipovány tak, že všechny THT součástky jsou umístěny ze spodní strany. Při používání jsou desky THT součástkami nahoru (SMD součástkami dolů) a jsou opatřeny distančními sloupky. Displeje jsou upevněny pomocí oboustranné lepicí pásky.

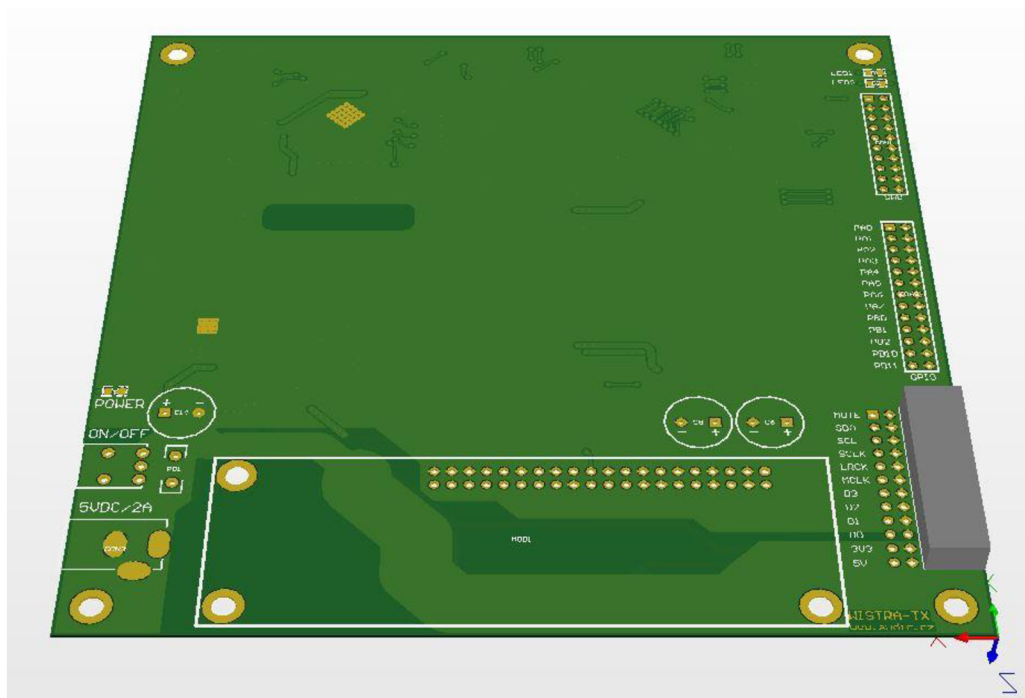
DPS vývojové desky vysílače a přijímače mají rozměry shodně 130 x 130 mm. Deska ADC má rozměry 55 x 45 mm. Deska DAC má pak 75 x 93 mm.

Vygenerované soubory obsahují kromě vrtání následující vrstvy: horní vrstvu, spodní vrstvu, horní masku, dolní masku, horní respektive dolní potisk, vrstvu frézy a vrstvu obrysu. Pro výrobu byla zvolena služba *Pool Service* poskytovaná firmou Gatema. Vodivé vrstvy jsou zobrazeny v příloze (B.1 až B.5).

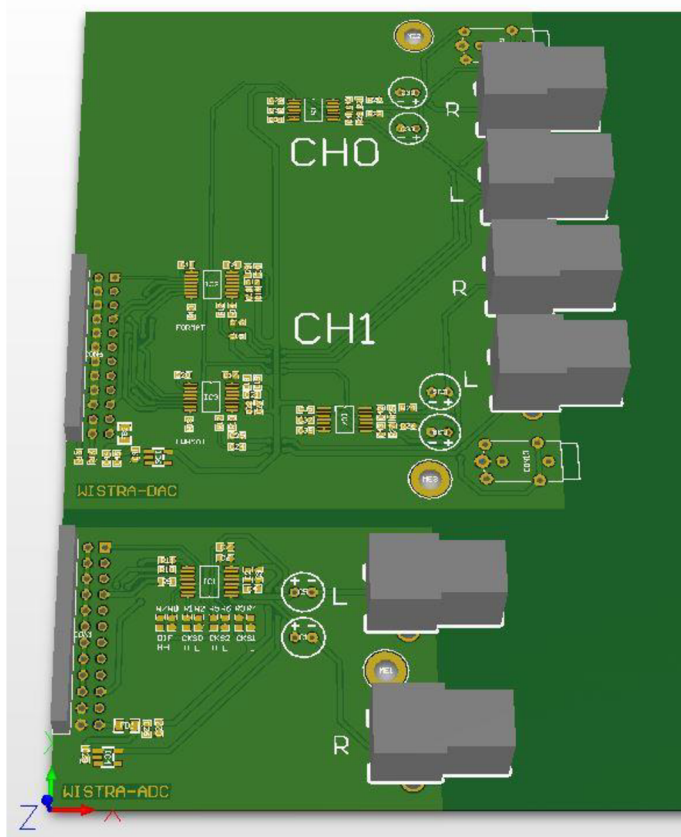
Výsledky jsou zobrazeny na následujících obrázcích (Obr. 8 až 17).



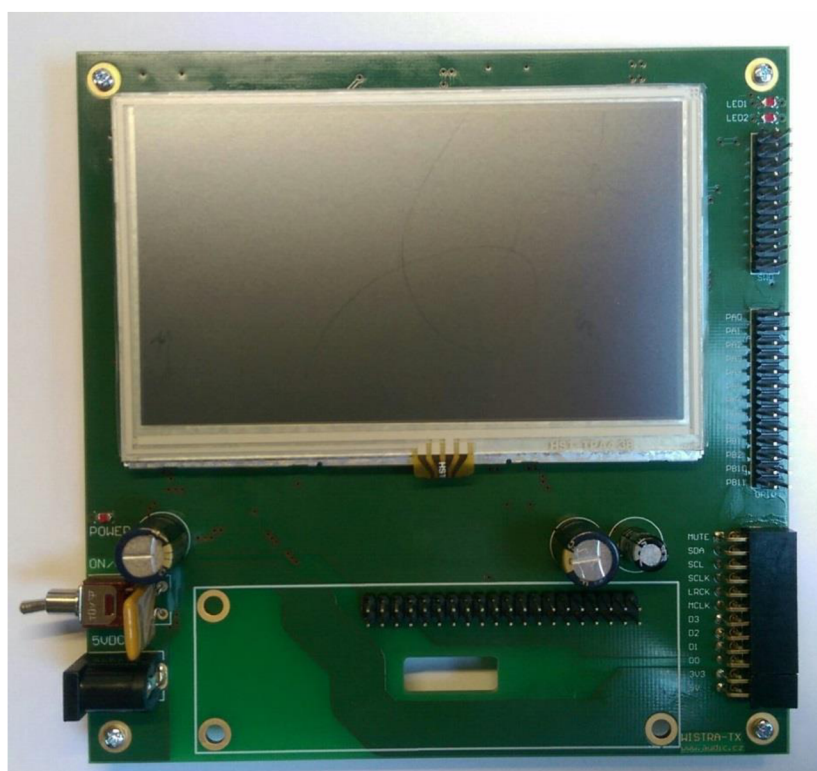
Obr. 8 3D zobrazení návrhu DPS vysílače shora



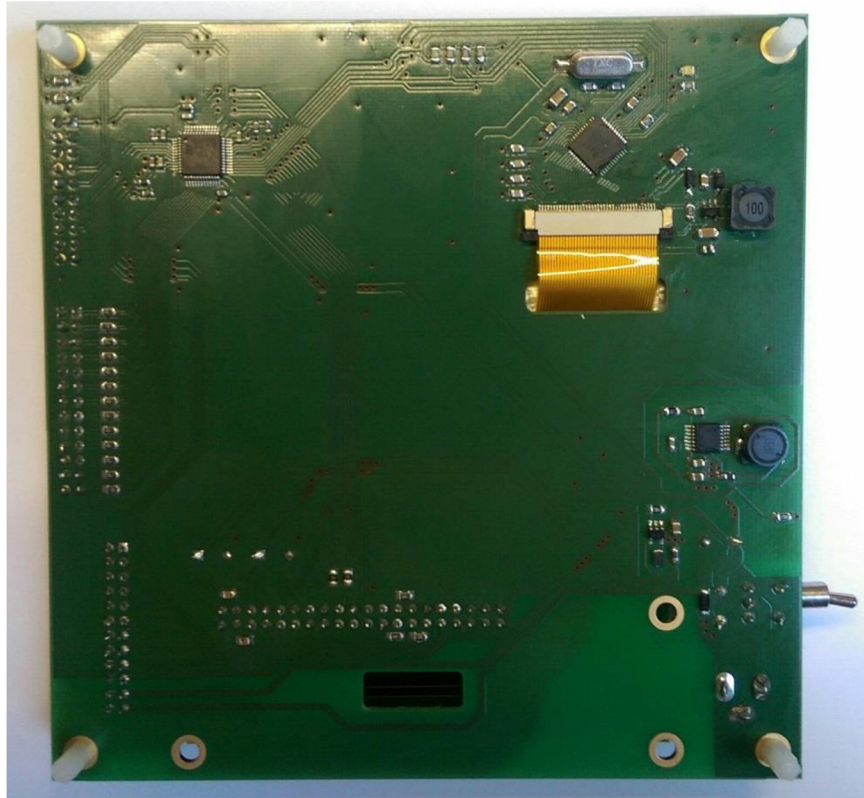
Obr. 9 3D zobrazení návrhu DPS vysílače zespodu



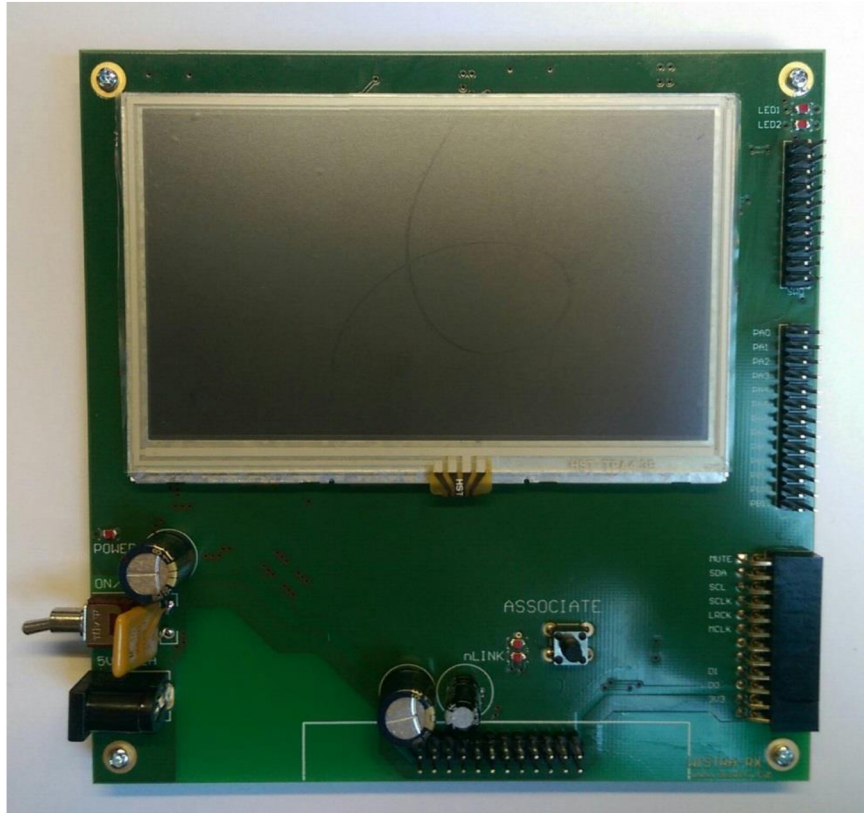
Obr. 10 3D zobrazení návrhu DPS AD a DA převodníku



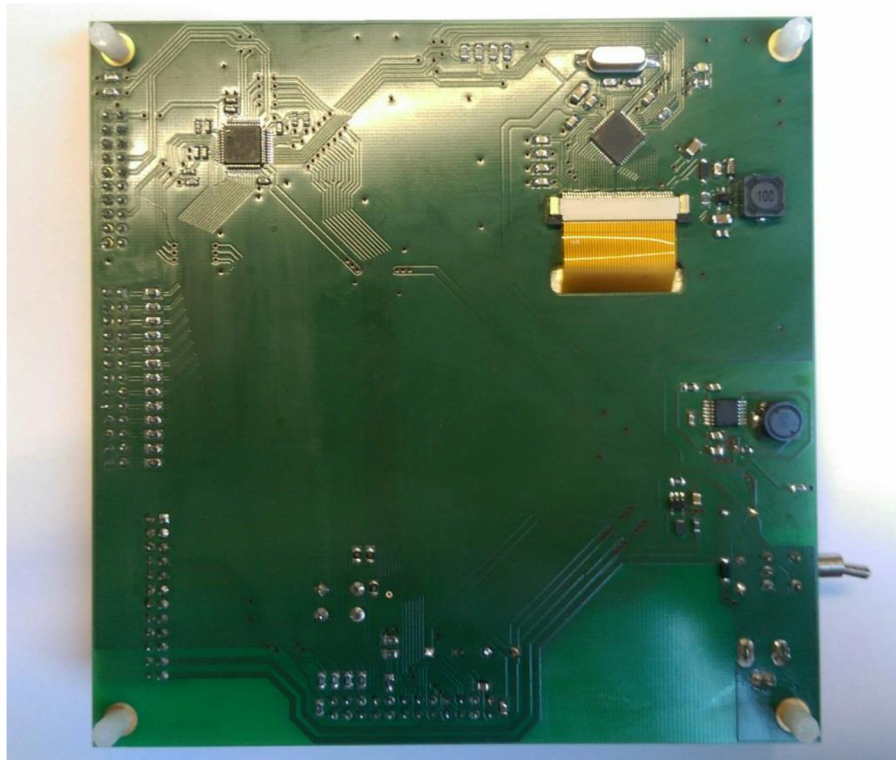
Obr. 11 Hotová vývojová deska vysílače – pohled shora



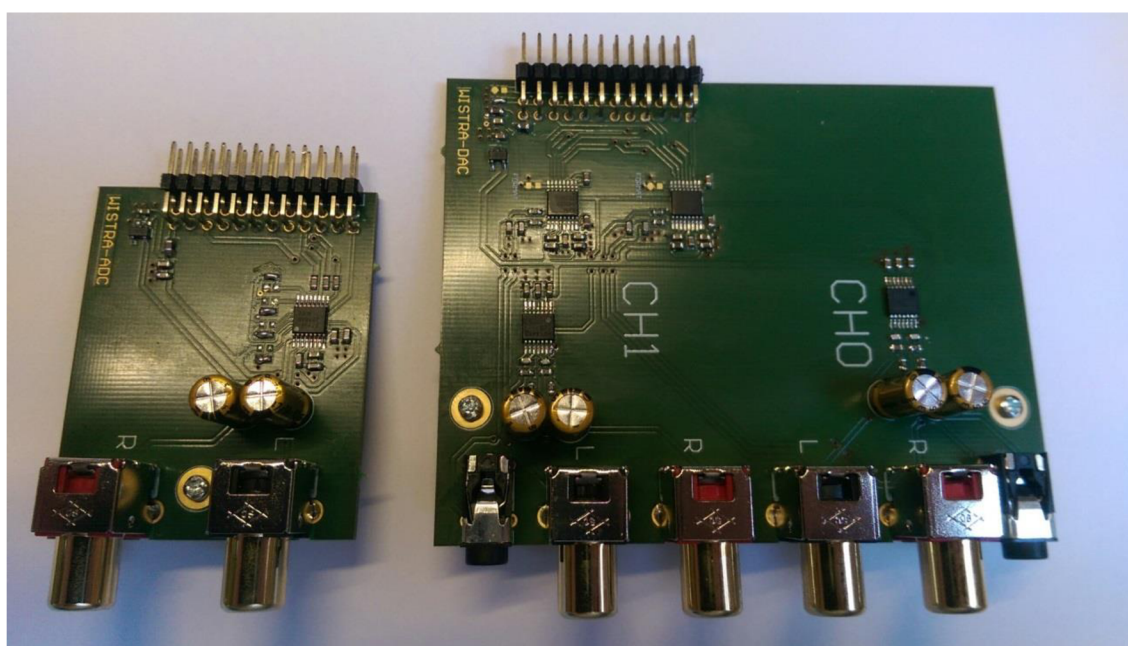
Obr. 12 Hotová vývojová deska vysílače - pohled zespoda



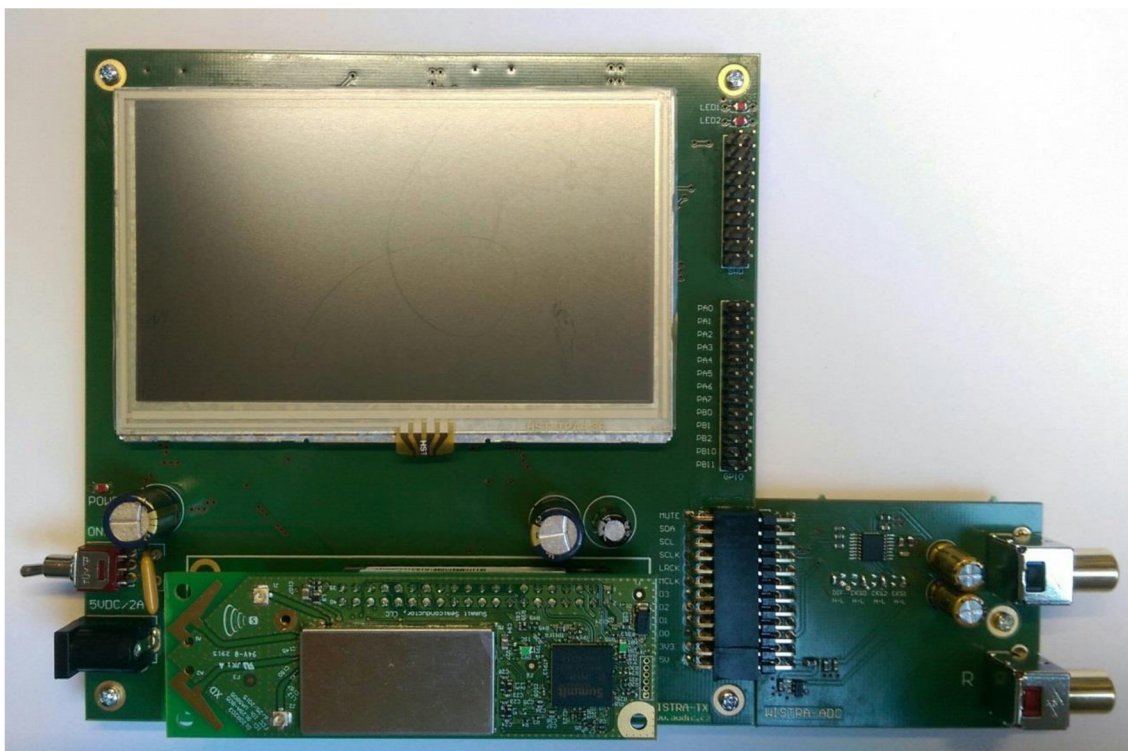
Obr. 13 Hotová vývojová deska přijímače - pohled shora



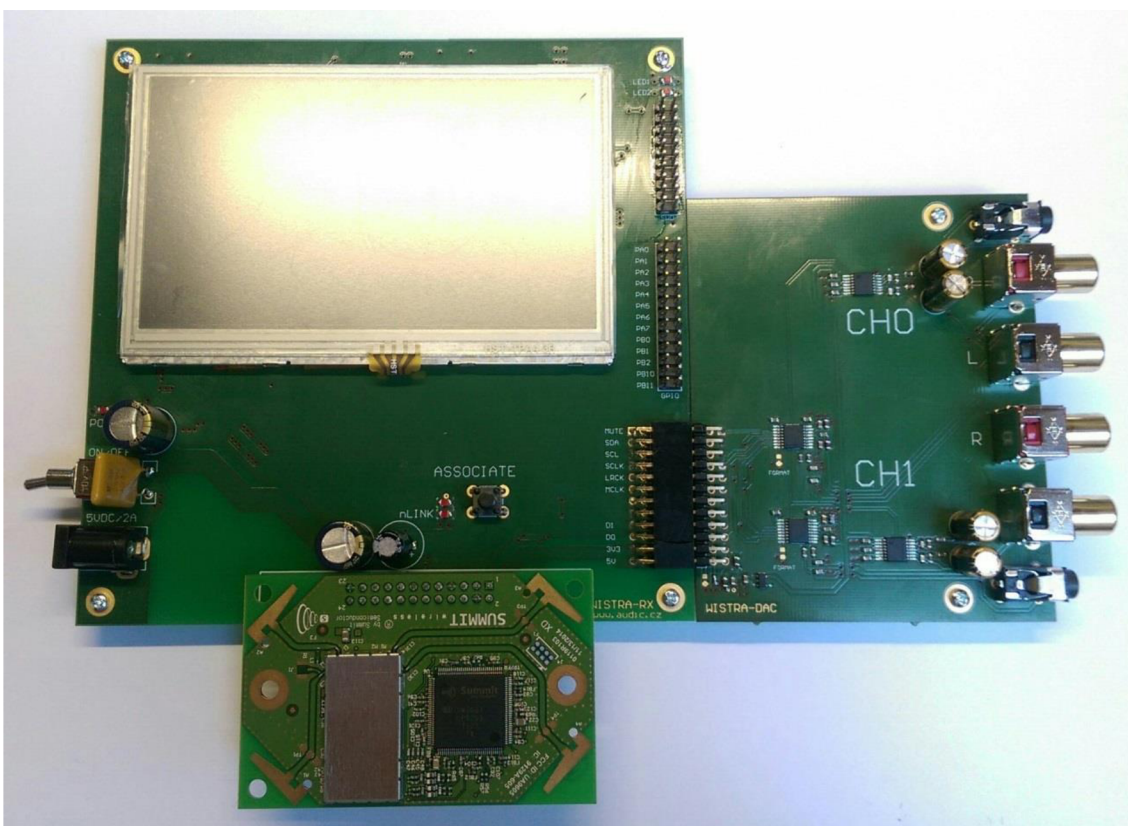
Obr. 14 Hotová vývojová deska přijímače - pohled zespoda



Obr. 15 Hotové DPS AD a DA převodníku



Obr. 16 Hotová sestava vysílače



Obr. 17 Hotová sestava přijímače

3 NÁVRH SOFTWARE

Následující kapitola dokumentuje navržený software. V první části je popsán ovladač pro grafický řadič FT800. Následuje popis struktury typické aplikace využívající tento ovladač. Poté je zde popsána vytvořená knihovna pro ukládání dat do flash paměti mikrokontroléru. Tuto kapitolu uzavírá popis aplikace pro vývoj a testování modulů pro přenos audio signálu.

3.1 Ovladač pro FT800

Cílem bylo vytvořit C ovladač (driver) umožňující plné využití možností obvodu. Dále pak vytvořit ukázkovou aplikaci sloužící jako kostru pro snadný návrh dalších aplikací. Tato kostra implementuje všechny potřebné soubory a funkce pro použití vlastních fontů a bitmap. Dále pak umožňuje změnu barevného schématu a jazyka použitého v aplikaci.

Ovladač je určen primárně pro FT800, je nicméně možné jeho použití pro kompatibilní čipy od firmy FTDI (FT801, FT810, FT811, FT812, FT813). Tyto řadiče jsou specifické tím, že umožňují tvorbu pokročilých GUI (*graphical user interface*) s minimálními nároky na řídicí mikrokontrolér. Obvod je možné připojit pomocí SPI (*serial peripheral interface*) s rychlostí až 30 MHz, nebo pomocí i2c (*Inter-Integrated Circuit*) do rychlosti 3.4MHz. Varianta FT800 umožňuje připojení displejů o rozlišení až 512×512 pixelů, přičemž každá barva je reprezentována 6-ti vodiči. Jde tedy o 18-bitovou hloubku (262144 barev). Řadič dále umožňuje generování PWM mono audio signálu a je tedy vhodný i pro pokročilá grafická rozhraní. Malé náročnosti pro řídicí mikrokontrolér je dosaženo mimo jiné pomocí widgetů, což jsou složitější prvky, jako například tlačítka, posuvníky, hodiny, stavové ukazatele a mnohé další.

Obvod dále podporuje připojení rezistivní dotykové vrstvy. Uživateli pak poskytuje nejen souřadnice stlačeného bodu, ale také informaci o objektu na daných souřadnicích. Uživatelská aplikace může každému vykreslenému prvku nebo objektu přiřadit tag, který slouží jako identifikátor objektu. Následným sledováním tagu může aplikace zjišťovat, zda došlo k dotyku objektu a libovolným způsobem reagovat.

Pro vykreslení obrazovky se vytváří DL (*Display-list*). Pro tento účel je vymezená paměť o velikosti 8192B, do které může mikrokontrolér zapisovat příkazy a následně je nechat vykreslit pomocí zápisu do dedikovaného registru. Do této paměti může ale zapisovat také grafický koprocessor, který vytváří widgety, a proto není vhodné do této paměti zapisovat přímo ale využívat raději FIFO koprocessoru k vykreslování, jak jednoduchých prvků, tak i widgetů. Tento blok paměti má velikost 4096B a pro jeho obsluhu má řadič dva registry sloužící jako zapisovací a čtecí ukazatel. Zapisovací ukazatel musí po zapsání příkazů inkrementovat aplikaci. Naopak čtecí ukazatel obsluhuje grafický koprocessor. Jsou-li si oba ukazatele rovné, znamená to, že všechny příkazy už byly zpracovány a zapsány do DL. FIFO je realizováno jako kruhový zásobník.

Výrobce poskytuje nástroje pro konverzi řetězců a bitmap do několika různých podporovaných formátů a nástroj pro konverzi audio signálu.

Dále je poskytován program generující kód pro tvorbu DL pomocí grafického rozhraní. Pro podobné účely lze použít také software Visual TFT od firmy Mikroelektronika.

Kromě zmíněných nástrojů existuje ovladač pro platformu arduino i ovladač v jazyce C poskytovaný formou example kódu. Tento ovladač má nicméně některé nedostatky, díky kterým klade vyšší nároky na použitý mikrokontrolér.

3.1.1 Závislost na hardwaru

Driver byl vyvíjen s cílem dosažení nezávislosti na platformě. Nicméně jeho použití pro GCC pro ARM a CodeVision pro 8-bitové mikrokontroléry Atmel (Microchip) vedlo k nutnosti vytvoření dvou verzí této knihovny a to zásluhou nestandardního chování kompilátoru CodeVision týkajícího se využívání některých klíčových slov (`const`, `flash`).

Pro použití v libovolné aplikaci je pak potřeba pouze správně nakonfigurovat soubor `FT800_library_cfg.h` (makra pro ovládání CS pro SPI, makra pro inicializaci SPI pro dvě různé rychlosti, definice datových typů, optimalizační konstantu, velikost interního pole pro zpracování řetězců, rozlišení a typ displeje, orientaci displeje...).

Pro použití pro jiný FTDI řadič je pak potřeba zkontrolovat a případně upravit nebo vyměnit soubor `FT800_library.h`, který obsahuje mimo jiné definice adres registrů a mapování paměti řadiče.

3.1.2 Struktura ovladače

Ovladač se skládá ze tří souborů:

- `FT800_library.c` - hlavní modul obsahující veškerý vykonávaný kód
- `FT800_library.h` - deklarace funkcí hlavního modulu, definice registrů, adresace paměti řadiče, definice parametrů pro jednoduché prvky a widgety
- `FT800_library_cfg.h` - konfigurace driveru specifická pro danou aplikaci a hardware

3.1.3 Hlavní modul

Modul obsahuje definice funkcí rozdělené do několika sekcí.

Rozdělení funkcí:

- zápis a čtení registrů (7 x)
- tvorbu a řízení DL (3 x)
- nastavení grafického kontextu (40 x)
- funkce pro grafický koprocessor (45 x)
- uživatelské funkce (19 x)

Sekce zápis a čtení registrů obsahuje funkce pro zápis a čtení pomocí SPI. Nutností je správné definování maker spojených se SPI (viz. 3.1.4).

Část pro tvorbu a řízení DL umožňuje přímé vytváření pro použití bez grafického koprocessoru a obsahuje funkci `dl_swap`, která vyvolá vykreslení hotového DL.

Sekce pro grafický kontext a koprocessor společně kompletně implementují API (*Application Programming Interface*) z dokumentu FT800 Series Programmer Guide, další sekce vznikaly postupně na základě potřeb během vývoje.

Funkce pro nastavení grafického kontextu mohou být použity přímo pro zápis do DL, nebo mohou být použity prostřednictvím FIFO koprocessoru, což je preferovaná varianta. Návrátová hodnota je vždy výsledkem logických operací mezi opcodem a parametry. Jednoduchým příkladem může být funkce `line_width`.

```
/* sets width of line primitive in 1/16th pixel precision */
uint32_t line_width(uint16_t width){
    uint32_t temp=0;
    temp=(width)|(0x0E000000);
    return temp;
}
```

Funkce pro grafický koprocessor zapisují do FIFO koprocessoru, ten následně vytváří DL. Tato sekce knihovny obsahuje kód pro veškeré widgety. Příkladem může být vykreslení tlačítka. Nejdříve je vždy kontrolován stav FIFO, poté následuje opcode a parametry specifické pro daný příkaz (například souřadnice, rozměry, vypisovaný řetězec,...), nakonec je inkrementován zapisovací ukazatel FIFO (včetně zarovnání na 4B).

Stav FIFO je kontrolován pomocí čtení registrů zapisovacího a čtecího ukazatele. Jejich srovnáním je možné zjistit, kolik je v kruhovém zásobníku místa. Jelikož jsou ale příkazy obvykle mnohem menší než je velikost FIFO, bylo definováno makro `BUFFER_RESERVE`. Tímto způsobem lze zamezit zbytečnému čekání mikrokontroléru na koprocessor FT800. Čekání tedy nastane pouze v případě, že je v paměti více nezpracovaných příkazů než odpovídá hodnotě `BUFFER_RESERVE`. Hodnota by neměla být příliš vysoká, protože nejdelší možný příkaz, který bude s jistotou správně zpracován je velikost FIFO – `BUFFER_RESERVE`. Pomocí měření počtu snímků za

vteřinu při různých vykreslovaných obrazovkách se ukázala dostačující hodnota 500, což zároveň v praxi neomezuje velikost nejdelšího příkazu (4096 – 500 = 3596), protože běžně používané příkazy zaberou maximálně několik set bajtů.

Dalším omezením je pak maximální velikost DL, která je pro FT800 8192B. Při použití widgetů je DL vytvářen koprocetorem a paměť zabraná v DL tak může být mnohonásobně větší než velikost příkazu zapsaná do paměti FIFO koprocetoru. Kontrola velikosti DL není v příkazech pro koprocetor implementována a je tedy zodpovědností aplikace toto kritérium dodržovat. V praxi je nicméně velikost paměti pro DL dostačující a její nedostatek hrozí jen při extrémně složitých vykreslovaných obrazovkách.

```

/* draw a button */
void cmd_button(sint16_t x, sint16_t y, sint16_t w, sint16_t h,
sint16_t font, uint16_t options, const char *s)
{
    uint32_t i=0;
    uint32_t read_addr;
    uint32_t write_addr;

/* Waiting is needed to ensure that the coprocessor buffer does not
overflow. BUFFER_RESERVE is used to make the program flow faster */
    uint8_t wait_condition;

    do{
        read_addr=rd16(REG_CMD_READ);
        write_addr=rd16(REG_CMD_WRITE);
        if(write_addr>=read_addr)
            wait_condition=
                ((write_addr-read_addr) >BUFFER_RESERVE?1:0);
        else
            wait_condition=
                ((write_addr+BUFFER_SIZE-read_addr)>BUFFER_RESERVE?
                1:0);
    }while(wait_condition);

    wr32(RAM_CMD+(write_addr), 0xFFFFFFFF);
    wr16(RAM_CMD+((write_addr+4)&(BUFFER_SIZE-1)), x);
    wr16(RAM_CMD+((write_addr+6)&(BUFFER_SIZE-1)), y);
    wr16(RAM_CMD+((write_addr+8)&(BUFFER_SIZE-1)), w);
    wr16(RAM_CMD+((write_addr+10)&(BUFFER_SIZE-1)), h);
    wr16(RAM_CMD+((write_addr+12)&(BUFFER_SIZE-1)), font);
    wr16(RAM_CMD+((write_addr+14)&(BUFFER_SIZE-1)), options);

    while(*(s+i)!=0){
        wr8(RAM_CMD+((write_addr+16+i)&(BUFFER_SIZE-1)), *(s+i));
        i++;
    }
    wr8(RAM_CMD+((write_addr+16+i)&(BUFFER_SIZE-1)), *(s+i));
    i++;
    wr16(REG_CMD_WRITE, (write_addr+16+i+3)&0xffc);
}

```

Sekce knihovny nazvaná uživatelské funkce se sice dále dělí na několik částí, nicméně obsahem všech jsou funkce, které souvisí s tvorbou GUI a jsou universálně použitelné pro různé projekty jako například:

- inicializace FT800
- inicializace tabulky barev
- inicializace tabulky řetězců
- inicializace bitmap a uživatelských fontů
- kalibrace dotykové vrstvy
- čtení tagu nebo souřadnic doteku
- konverze řetězců mezi kódováním překladače a kódováním uživatelského fontu
- inteligentní zalamování řetězců
- počítadlo FPS
- ...

3.1.4 Konfigurace ovladače

Konfigurace driveru pro použití pro jiný kompatibilní řadič (FT801, FT810, FT811, FT812, FT813) je možná pomocí úprav souboru `FT800_library.h`.

Veškerá konfigurace pro konkrétní hardware a konkrétní aplikace je pak prováděna uvnitř souboru `FT800_library_cfg.h`. Pro tento účel je nutné definování následujících maker:

- `CS_ON`, `CS_OFF` – ovládání pinu *chip select* pro SPI
- `PD_ON`, `PD_OFF` – ovládání *power down* pinu řadiče
- `SPI_INIT_SLOW`, `SPI_INIT_FAST` – inicializace periferie SPI (do 10MHz a do 30MHz)
- `SPI_MASTER_TX_RX (c)` – odeslání a současné přijetí 1B po SPI
- `DELAY_MS (ms)` – zpoždění nutné pro bezpečnou inicializaci řadiče
- `WATCHDOG_RESET` – resetování obvodu *watchdog* v případě, že je použit
- `uint32_t`, `uint16_t`, `sint16_t`, `uint8_t` – typy závislé na platformě
- `BUFFER_RESERVE` – optimalizační parametr
- `MAX_PARSECHAR_NUMBER` – omezení pro zalamování řetězců
- `MAX_STRING_LEN` – velikost pole pro práci s řetězcí
- `DISP_HEIGHT`, `DISP_WIDTH` – rozlišení použitého displeje
- `DISP_REG_XXX` – sada 11-ti parametrů specifických pro použitý displej (například polarita hodin, vertikální a horizontální synchronizace,...)

3.1.5 Struktura aplikací

Driver byl použit již pro několik aplikací různé složitosti. Pro vytvoření nové aplikace s pokročilým graficko uživatelským rozhraním stačí do projektu vložit několik souborů a pak už jen vytvářet jednotlivé obrazovky. Kostra nového projektu je tedy daná a je výsledkem postupného vývoje. Potřebné soubory:

- `ftf.c`, `ftf.h`, `tft_cfg.h` – hlavní modul ve kterém se nachází jednotlivé obrazovky a konfigurace projektu
- `FT800_library.c`, `FT800_library.h`, `FT800_library_cfg.h` – ovladač grafického řadiče a jeho konfigurace
- `tft_global_variables.h` – soubor pro definici globálních proměnných `tft.c`
- `strings.h` – soubor s texty s tabulkou konstantních řetězců pro všechny podporované jazyky
- `font_data.h` – soubor obsahující data pro uživatelské fonty
- `bitmaps.h` – soubor obsahující bitmapy
- `tft_types.h` – definice konfigurační struktury (jazyk, podsvit, barevné schéma, další parametry specifické pro aplikaci)

Aplikace je strukturována jako stavový automat se stavovou proměnnou `scr`. Jednotlivé obrazovky často přistupují ke stejným datům a většina proměnných je tak pro modul `tft.c` globálních. Během běhu programu je potřeba periodicky provádět některé operace jako například kontrolu stavů časovačů nebo čtení tagu. Tyto operace jsou prováděny mezi vykreslením obrazovek z hlavní smyčky, která obrazovky volá. Obrazovky („screeny“) jsou tedy implementovány jako jednorázové vykreslení, ne jako smyčky. Hlavní smyčka pak může být realizována následujícím způsobem.

```
while(1){
    /* Touch handling */
    ...
    /* Timers handling */
    ...
    /* Screen state machine */
    switch (scr) {
        case MAIN_SCREEN: //MAIN screen
            main_screen();
            break;
        case SETTINGS_SCREEN: //settings
            settings_screen();
            break;
        case PASSWORD_SCREEN: //password
            password_screen();
            break;
        case
            ...
    }
}
```

Každé překreslení obrazovky je reprezentováno vykreslením jednoho displej listu (DL). Každý DL tvořen grafickým koprocesorem začíná voláním `cmd_dlstart`, dále pokračuje vykreslováním dalších příkazů s případnými výpočty. Ukončuje ho příkaz `cmd(display())` a vykreslen je pak pomocí `cmd_swap()`.

Driver je koncipován bez vlastního zásobníku, tj. příkazy se posílají do FT800 přímo a mezi jednotlivými příkazy mohou být prováděny výpočty (např. souřadnic), díky čemuž je tento přístup výpočetně optimálnější než při použití zásobníku. Navíc by bylo nutné alokovat značný prostor v SRAM (pro FT800 až 4KB), což by omezilo použití driveru pro menší mikrokontroléry. Obrazovky jsou vykresleny typicky s 20 až 70 FPS podle jejich složitosti. Zjednodušený příklad může vypadat následujícím způsobem.

```
void screen_example(void){
    /* Start new DL */
    cmd_dlstart();
    /* Background gradient */
    cmd_gradient_custom(COLOR_GR1);
    /* RGB type of color for the next comments */
    cmd_set_color(COLOR_STR3, COLOR_RGB);
    /* Print text to the center of the screen */
    cmd_text(DISPLAY_WIDTH/2, DISPLAY_HEIGHT/2, 28, OPT_CENTERX, "Hello World");
    /* Draw button with tag 2, if it is currently touched it is
    drawn as flat */
    cmd(tag(2));
    cmd_button(5.5, 20, 30, (atag==2)*OPT_FLAT, "Press here");
    .
    .
    .
    /* End of DL */
    cmd(display());
    /* Draw */
    cmd_swap();
    /* Touch handling */
    switch(rtag){
        case 2:
            scr = SCREEN_EXAMPLE2;
            break;
    }
}
```

FT800 má integrovány dva fonty, každý v několika velikostech. Pokud aplikace vyžaduje font například s českou diakritikou, je možné si tento font pomocí poskytovaného nástroje pro příkazovou řádku vygenerovat. K tomu je potřeba soubor fontu a textový soubor se znaky, které mají být vytvořeny. Výsledná data lze vložit například jako hlavičkový soubor do kódu. Dále je potřeba font do řadiče nahrát a správně jej inicializovat. K tomuto účelu slouží uživatelská funkce `custom_font_init`. Aby bylo možné font použít, je nejdříve nutné konvertovat řetězce na správné kódování (znaky ve vytvořeném fontu jsou kódovány podle pořadí v textovém souboru před konverzí, kdežto řetězce z kódu jsou kódovány podle použitého editoru a kompilátoru). K tomu slouží další uživatelská funkce obsahující převod znaků podle kódovací tabulky `encoding_convert`.

Podobně jako s fonty je možné dedikovaným nástrojem konvertovat bitmapy

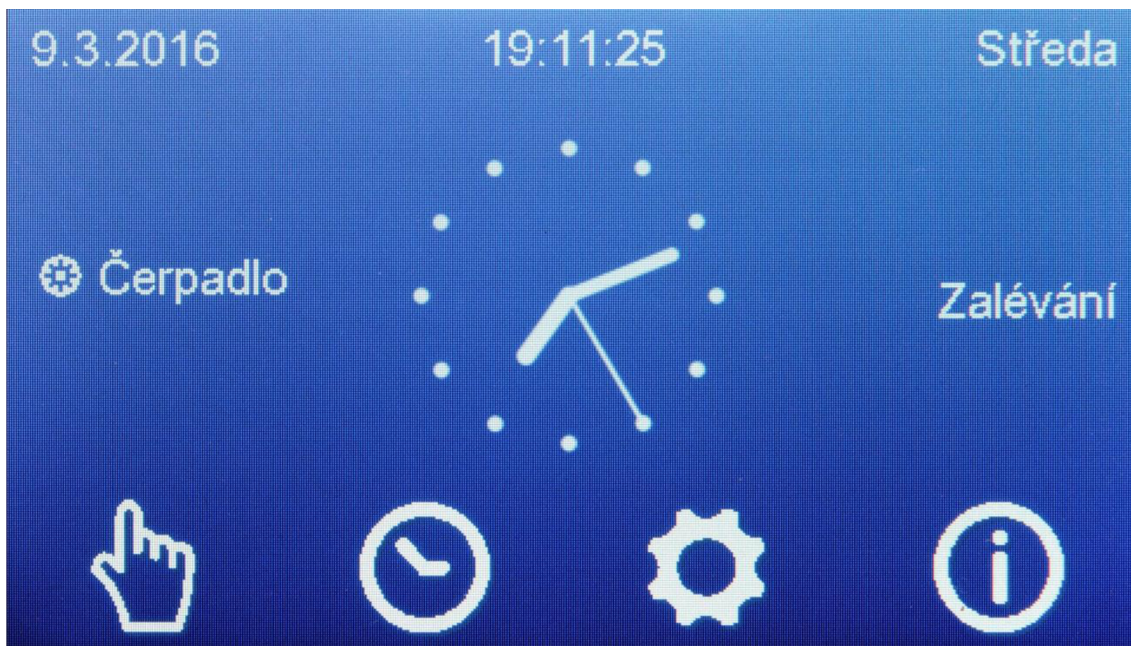
v různých formátech. Ty je následně potřeba, stejně jako fonty, nahrát do paměti a inicializovat (adresa, rozměry, handle, formát, ...).

Aby bylo možné za běhu měnit jazyk aplikace, obsahuje program vícerozměrné pole s adresami konstantních řetězců. První index určuje typ textu (např. „Nastavení“, „Vytvořit položku“,...), druhý index vybírá jazyk. Zjednodušený přístup k těmto řetězcům pak poskytuje uživatelská funkce `strings`.

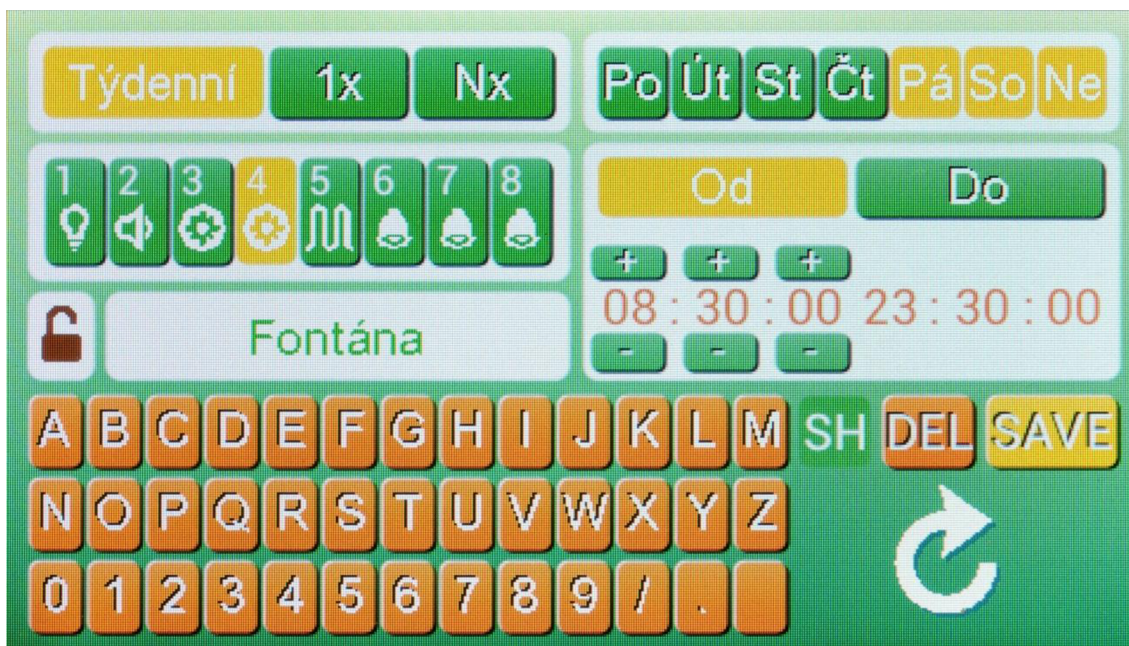
Podobně jako v případě řetězců obsahuje aplikace tabulku barev. První index určuje tzv. barevnou třídu (například tlačítko typu 1 má vytvořenou třídu BUT1, text typu 1 třídu STR1, bitmapy popředí třídu BMFG, gradient na pozadí třídu GR11, atd.), druhý index `par` určuje složku RGB. Pro změnu barevného schématu stačí předat driveru adresu jiné tabulky barev. Jelikož grafický kontext používá tři druhy barev (`COLOR_RGB`, `FGCOLOR`, `BGCOLOR`), byla vytvořena funkce `cmd_set_color`, která nastavování zjednodušuje.

3.1.6 Příklady GUI

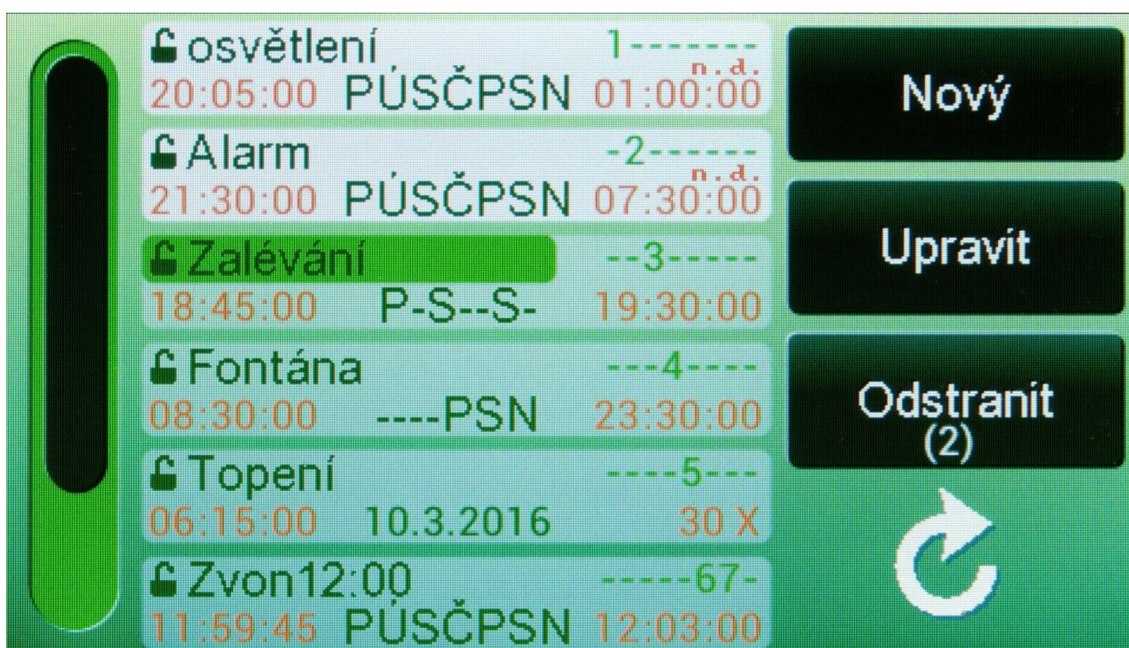
Tato kapitola obsahuje ukázky obrazovek z některých GUI vytvořených pomocí popsaného ovladače. Tyto aplikace jsou vždy strukturovány výše popsaným způsobem (viz. 3.1.5). Tato fotodokumentace (Obr. 18 až 20) souží pouze pro ukázání některých možností grafického řadiče FT800 a vytvořeného ovladače. Obrázky byly převzaty ze stránek zadávající firmy [12].



Obr. 18 Příklad použití ovladače (widget hodiny, použití uživatelského fontu, bitmapy)



Obr. 19 Příklad použití ovladače (widgety tlačítek a klávesnice)



Obr. 20 Příklad použití ovladače (widget posuvníku)

3.2 Knihovna pro využití paměti FLASH

Vzhledem k tomu, že pro vývojové desky byly zvoleny mikrokontroléry od firmy ST Microelectronic, které převážně nemají integrovanou paměť EEPROM, byla vytvořena knihovna umožňující ukládání trvalých dat do paměti FLASH.

3.2.1 Princip činnosti

Na rozdíl od paměti EEPROM nelze paměť typu FLASH libovolně přepisovat. Aby mohl být proveden zápis, je nejdříve nutné paměť odemknout pro zápis, poté musí být vymazána celá stránka (*page*) a teprve pak je možná provést zápis. Poté by měla být paměť znovu uzamčena. Velikost jedné stránky je typicky 1 KB nebo 2KB. [8]

Vzhledem k tomu, že výrobce garantuje pouze 1000 cyklů paměti a změna jediného bajtu by znamenala přepsání celé stránky, je pro praktické použití nutné proces optimalizovat. Na každou adresu lze zapsat po vymazání stránky pouze jednou. [8]

Prvním krokem je využití více stránek, které jsou využívány cyklicky, čímž je zátěž rovnoměrně rozložena. S tímto přístupem však nastává nutnost přidávat ke každé stránce identifikační údaj potřebný pro nalezení poslední použité stránky po startu systému. Pro tento účel by mohlo posloužit počítadlo s maximální hodnotou větší než je počet využívaných stránek. Současně by muselo být kontrolováno přetečení počítadla.

Jelikož je ale běžné že aplikace využívá jen malé množství nevolatilní paměti, je možné proces ještě více zoptimalizovat a to tak, že každá stránka bude využita několikrát před přepnutím na stránku následující. Například pokud by byl využíván mikrokontrolér s pamětí o velikosti stránky 1 KB a aplikace potřebovala uchovávat 50 B dat, bylo by možné životnost paměti až 19 krát zvětšit. Na začátku každé stránky musí být alokováno tolik adresovatelných buněk (mikrokontroléry ST umožňují adresaci půl slov tj. 2B), kolikrát bude maximálně možné do stránky zapsat před přesunutím na stránku další. Při každém zápisu dat na stránku je zapsáno do další buňky. Matematické řešení problému pro tyto mikrokontroléry pak ukazuje následující vztah (1):

$$n = \frac{PAGE_SIZE - m * 2B}{data_size} \quad (1)$$

kde n udává, kolikrát může být stránka použita před přesunem na stránku další, m určuje velikost hlavičky stránky a $data_size$ je celková velikost dat, která aplikace ve FLASH uchovává. Nejúspěšnější případ nastane tehdy, když hlavička stránky bude co nejmenší. Tento předpoklad je popsán rovnicí (2). Optimální řešení pak nastane po dosazení rovnice 2 do rovnice 1. Výsledný vztah popisuje rovnicí (3):

$$n = m \quad (2)$$

$$n = \frac{PAGE_SIZE}{data_size + 2B} \quad (3)$$

a tento vztah tak potvrzuje předchozí výpočet. Tedy počítáno celočíselně, že $1024/(50+2) = 19$.

Knihovna je pak implementována s pomocí kombinace obou výše popsaných principů. Současně uvažuje stavy, při kterých by mohlo dojít ke ztrátě dat při výpadku napájení, při zápisu na stránku nebo při přechodu na další stránku.

3.2.2 Použití knihovny

Knihovna se skládá ze souborů `non_volatile_variables.c` a `non_volatile_variables.h`. Před použitím je nutné nejdříve ve zmíněném hlavičkovém souboru definovat následující makra:

- `FLASH_BLOCK_PAGE_NUM` – počet stránek které mají být využívány
- `PAGE_STATUS_SIZE` – udává, kolikrát může být každá stránka použita před přepnutím na stránku další (odpovídá m viz. 3.2.1)
- `FLASH_BLOCK_START` – adresa první stránky
- `PAGE_SIZE` – velikost stránky

Dále je vhodné upravit pravidla pro linkování (*linker-script*) tak, aby bylo zajištěno, že využitím knihovny nedojte k přepsání programu. Toho lze docílit například definováním nového bloku paměti, do které nejsou směrovány žádné sekce programu. Příklad může být následující úsek kódu.

```
MEMORY
{
    RAM (xrw)      : ORIGIN = 0x20000000, LENGTH = 32K
    FLASH (rx)     : ORIGIN = 0x80000000, LENGTH = 216K
    MY_FLASH (rx)  : ORIGIN = 0x08036000, LENGTH = 40K
}
```

Přičemž blok `MY_FLASH` již ve zbytku *linker-scriptu* nefiguruje.

Knihovna definuje strukturu s názvem `NV`, která je aplikaci dostupná jako externí po vložení hlavičkového souboru. Obsah této struktury je pak specifický pro každou aplikaci (uživatel knihovny zde definuje libovolné proměnné). Struktura je z FLASH paměti nahrána do RAM pomocí volání `NV_get()` a po provedení změn je uložena pomocí volání `NV_set()`.

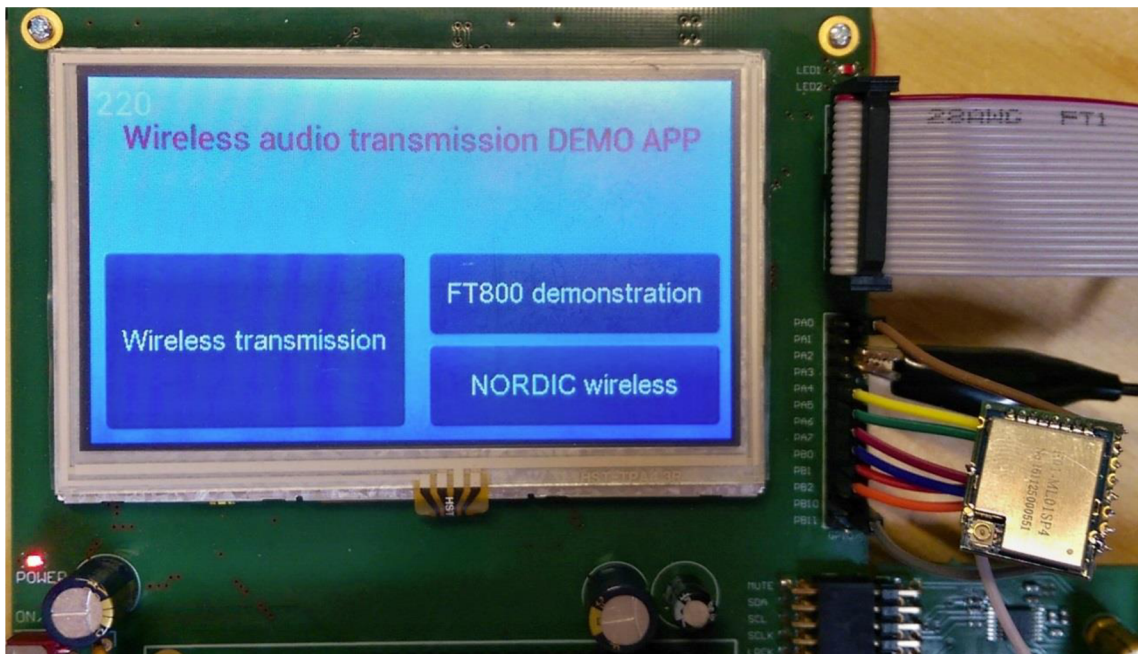
3.2.3 Shrnutí možností knihovny

Tato knihovna umožňuje pohodlnou práci s nevolatilní pamětí pomocí globální struktury `NV`, současně řeší problém s omezenou životností FLASH paměti pomocí střídáním stránek a vícenásobnému používání každé ze stránek. Přičemž veškeré tyto úkony jsou řešeny vnitřně a cílem je dosažení maximální bezpečnosti dat. Uživatel musí pouze správně nastavit velikost stránky dané mikrokontrolérem, dále adresu začátku a počet stránek tak, aby nebyla adresována neexistující paměť. Kromě toho má uživatel možnost optimalizovat využití paměti vhodnou volbou maxima opakovaného využití stránky. Jediným omezením knihovny je velikost jedné stránky (`PAGE_SIZE`). Celková velikost ukládaných dat musí být menší než tato hodnota. Velikost stránky je nicméně typicky 1 KB nebo 2 KB, což odpovídá velikostem běžně integrovaných pamětí EEPROM.

3.3 Ukázková aplikace

V rámci této práce byla vytvořena demonstrační aplikace plnící několik cílů. Prvním cílem bylo samotné oživení a otestování Summit modulů určených pro přenos audio signálu ve standardu WiSA. Dalším cílem pak bylo demonstrování některých možností grafického řadiče FT800 a funkcionality spojené s navrženým programem (viz. 3.1.5).

Po spuštění zařízení a inicializaci systému se objeví obrazovka s tlačítky „Wireless transmission“, „FT800 demonstration“ a „NORDIC wireless“. Stisknutí tlačítka vyvolá přesun do daného menu. Poslední zmíněné menu bylo přidáno nad rámec této práce pro účely portování a testování driveru pro moduly s čipem nfr24l01 od firmy Nordic pro bezdrátový přenos dat v pásmu 2.4GHz. Úvodní menu je zachyceno na následujícím obrázku (Obr. 21).

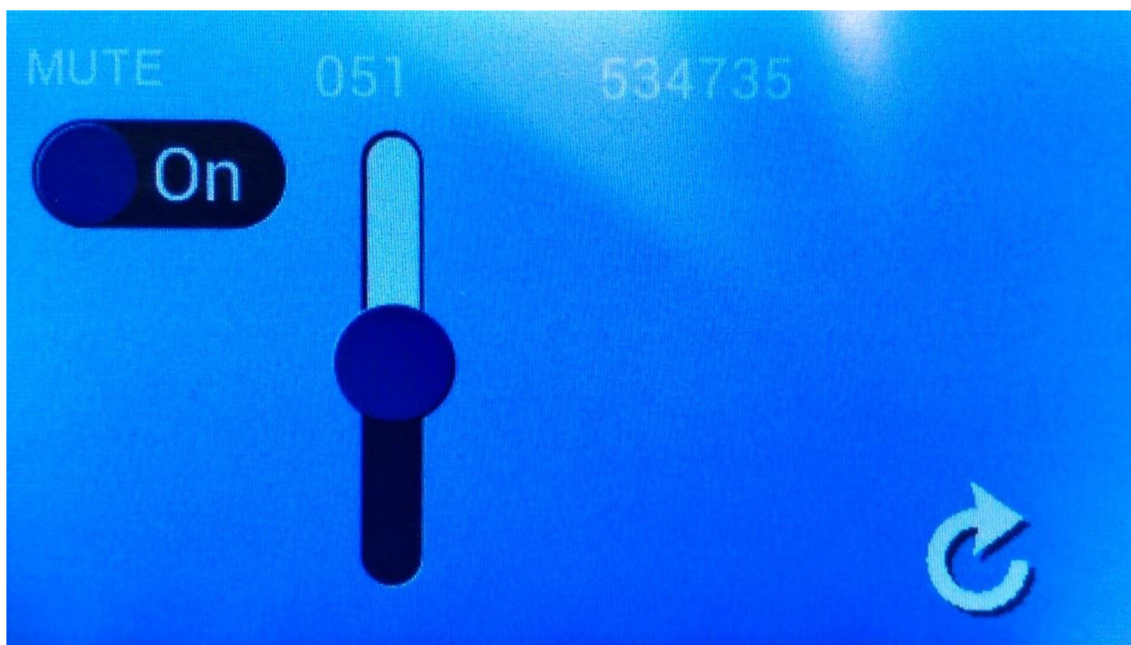


Obr. 21 Vstupní obrazovka ukázkové aplikace

3.3.1 Menu Wireless transmission

Z tohoto menu je ovládána veškerá funkcionálníta používaná pro usnadnění vývoje softwaru a nastavení modulů. Je zde přepínač ovládající pin pro zapnutí či vypnutí modulu a tlačítka pro inicializaci modulu. Dále jsou zde umístěna tlačítka pro vstup do podmenu, která pak umožňují měření teploty čipu, změnu hlasitosti nebo ovládání pinu MUTE pro úplné vypnutí audio cesty. Toto menu je zachyceno na následující obrázku (Obr. 22).

Jelikož při přenosu audio signálu dochází k občasnému nežádoucímu rozpojení přenosu a opětovné synchronizaci, není ještě vývoj zcela ukončen a tudíž i toto menu se může v budoucnu měnit či rozšiřovat podle povahy problému. Řešení tohoto nedostatku probíhá ve spojení přímo s výrobcem modulů (Summit Wireless). Informace obsažené v této kapitole (3.3.1) jsou vztažené k datu odevzdání práce.

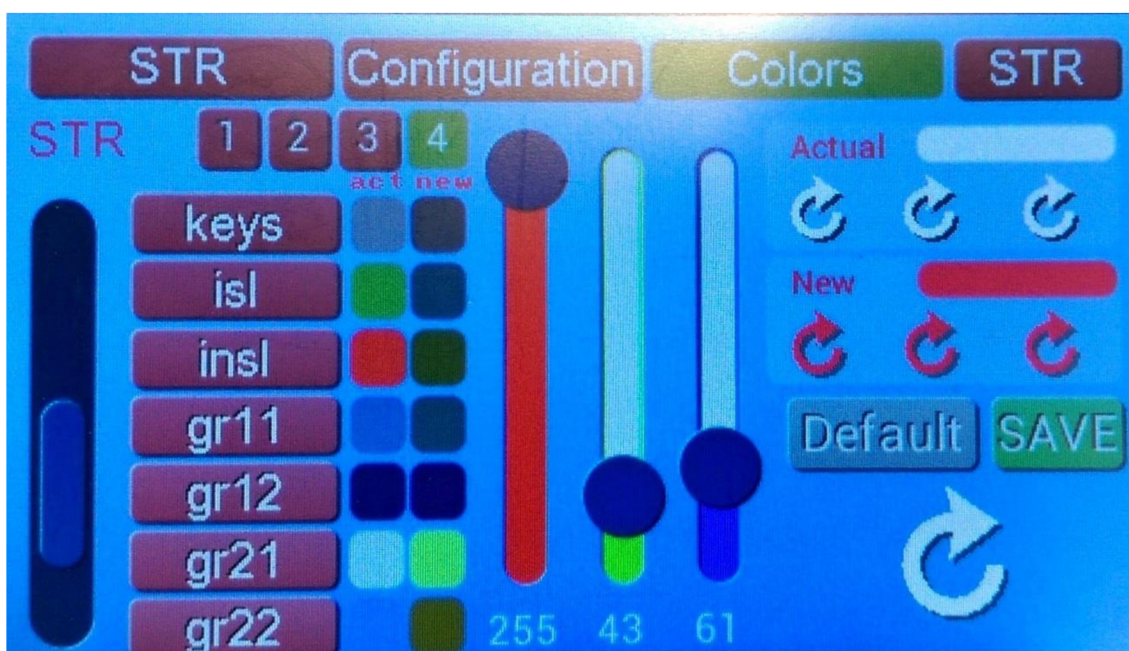


Obr. 22 Obrazovka pro nastavení hlasitosti

3.3.2 Menu FT800 demonstration

Toto menu slouží čistě pro demonstrační účely. Do několika záložek jsou zde rozmístěny různé widgety.

Je zde demonstrováno psaní textů na klávesnici s uživatelským fontem obsahujícím české a slovenské znaky. Dále je zde záložka posuvníky pro tvorbu barevných schémat (viz Obr. 23). Menu bylo navrženo tak, aby uživatel mohl měnit RGB složky jednotlivých barevných tříd a vždy viděl příklad použití třídy jak v aktuální tak nové barevné konfiguraci. Další záložka pak může sloužit k libovolné konfiguraci parametrů, jejichž funkcionality není v této demonstrační aplikaci určena. Poslední záložka pak obsahuje tlačítko pro obnovení továrního nastavení a tlačítko pro vyvolání kalibrace dotykového ovládání.



Obr. 23 Obrazovka pro nastavení barev

ZÁVĚR

Práce popisuje návrh hardwaru vývojových desek umožňujících přenos audio signálu ve standardu WiSA. Během vývoje zařízení se podařilo splnit všechny vytyčené cíle. Zařízení jsou opatřena dotykovými displeji, umožňují připojení externích současně vyvinutých DPS audio převodníků a mají vyvedená GPIO pro libovolné použití ve vyvíjené aplikaci.

Byla navržena obsáhlá knihovna pro obsluhu grafického řadiče FT800 (a kompatibilních verzí tohoto obvodu) pokrývající veškerou jeho funkcionalitu. Dále byla vytvořena knihovna pro zjednodušenou práci s pamětí FLASH pro mikrokontroléry s architekturou od firmy ST Microelectronics. Tato knihovna umožňuje využití paměti FLASH pro ukládání dat, čímž nahrazuje u většiny mikrokontrolérů této firmy scházející paměť typu EEPROM. Tato knihovna je optimalizována na dosažení maximální životnosti paměti a jednoduché použití v aplikaci. Spolu s vytvořenými knihovnami byla navržena architektura pro aplikace s použitým grafickým řadičem umožňující snadnou tvorbu pokročilých graficko uživatelských rozhraní.

Pro vývoj a demonstraci byla vytvořena ukázková aplikace využívající výsledky předchozí práce.

Vývojové desky již byly použity pro portování ovladačů a testování modulů pro bezdrátový přenos dat nrf24l01.

Podařilo se uskutečnit přenos audio signálu ve vysoké kvalitě ve standardu WiSA. Vývoj nebyl k datu odevzdání práce zcela ukončen, protože při přenosu dochází k občasnému rozpadu spojení a opětovné synchronizaci. Tento nedostatek je řešen ve spolupráci s výrobcem použitých modulů. Druhým nedostatkem je šíření rušení po napájení od vysílacího modulu (2KHz, 20mVpp), které v původním návrhu není zcela potlačena i přes dodržení veškerých požadavků daných dokumentací. Toto rušení se může superponovat ke zvukovému signálu a způsobit slyšitelné pískání. Tento problém může být řešen přidavnými blokovacími kondenzátory vhodné velikosti s případným přidáním filtru k napájení ADC převodníku.

LITERATURA

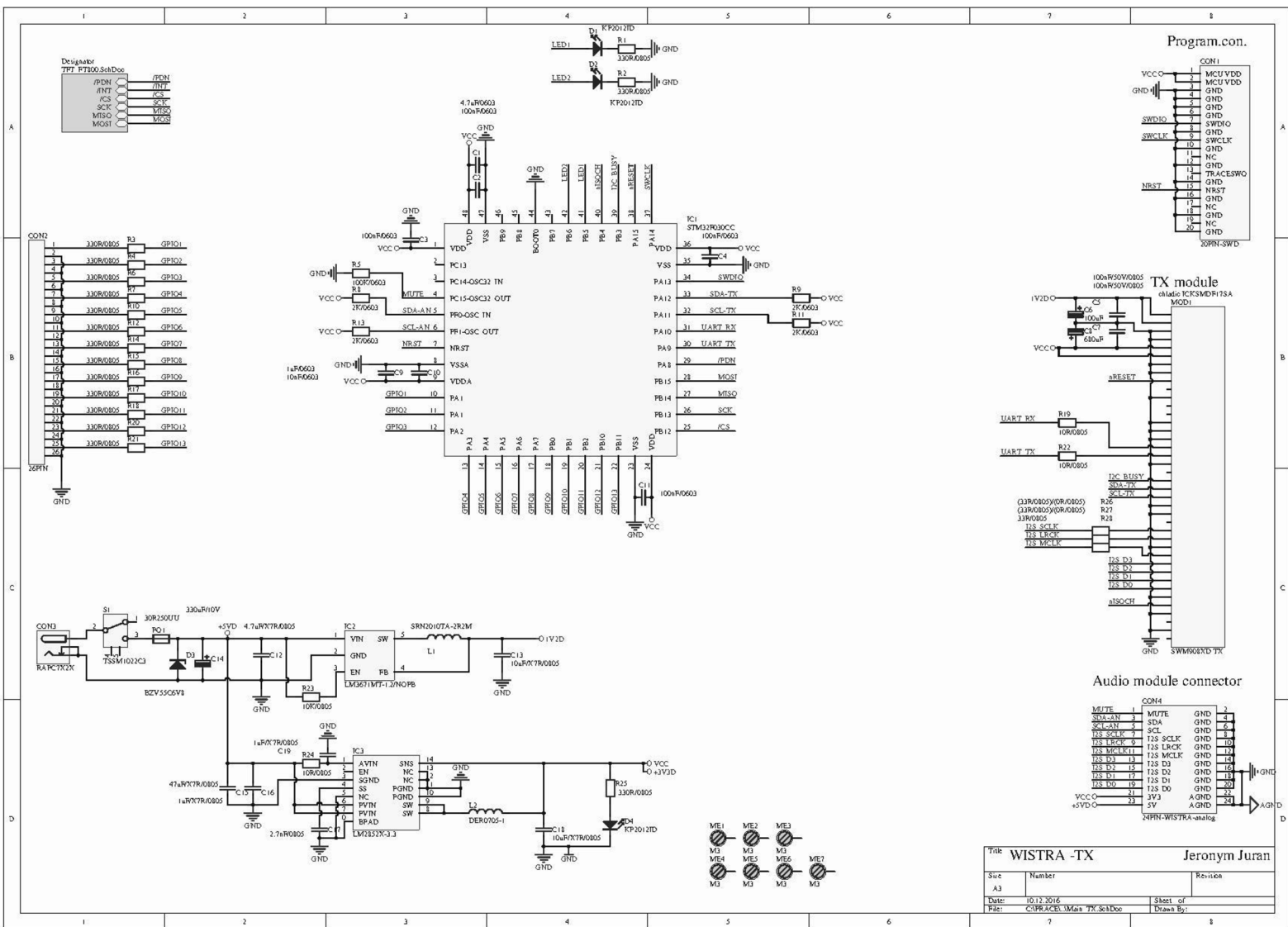
- [1] Digital audio broadcasting. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation [cit. 2016-12-10]. Dostupné z: https://en.wikipedia.org/wiki/Digital_audio_broadcasting
- [2] Wireless Speaker and Audio (WiSA™) Association White Paper. *Homeautomationinfo* [online]. [cit. 2016-12-10]. Dostupné z: http://www.homeautomationinfo.com/Drupal/files/WiSA_whitepaper_0809.pdf
- [3] Summit Transmitter Extended Distance. Summitwireless [online]. [cit. 2016-12-10]. Dostupné z: <http://www.summitwireless.com/summit-tx-xd.html>
- [4] Summit SWM908 XD TX Module. Summitwireless [online]. [cit. 2016-12-10]. Dostupné z: http://www.summitwireless.com/uploads/6/5/9/2/65922691/summit_xd_tx_mod_pb_v1.7.pdf
- [5] Summit Receiver Extended Distance. Summitwireless [online]. [cit. 2016-12-10]. Dostupné z: <http://www.summitwireless.com/summit-rx-xd.html>
- [6] Summit SWS997 XD RX Module. Summitwireless [online]. [cit. 2016-12-10]. Dostupné z: http://www.summitwireless.com/uploads/6/5/9/2/65922691/summit_xd_rx_mod_pb_v1.4.pdf
- [7] FT800. FTDI [online]. [cit. 2016-12-10]. Dostupné z: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT800.pdf
- [8] STM32F030x4 STM32F030x6 STM32F030x8 STM32F030xC. St [online]. [cit. 2016-12-10]. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/datasheet/a4/5d/0b/0e/87/c4/4d/71/DM00088500.pdf/files/DM00088500.pdf/jcr:content/translations/en.DM00088500.pdf>
- [9] AK5386. Akm [online]. [cit. 2016-12-10]. Dostupné z: <https://www.akm.com/akm/en/file/datasheet/AK5386VT.pdf>
- [10] AK4430. Akm [online]. [cit. 2016-12-10]. Dostupné z: <https://www.akm.com/akm/en/file/datasheet/AK4430ET.pdf>
- [11] MAX4410. Maximintegrated [online]. [cit. 2016-12-10]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/MAX4410.pdf>
- [12] AUDIC. *AUDIC* [online]. [cit. 2017-04-16]. Dostupné z: <https://www.audic.cz/>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

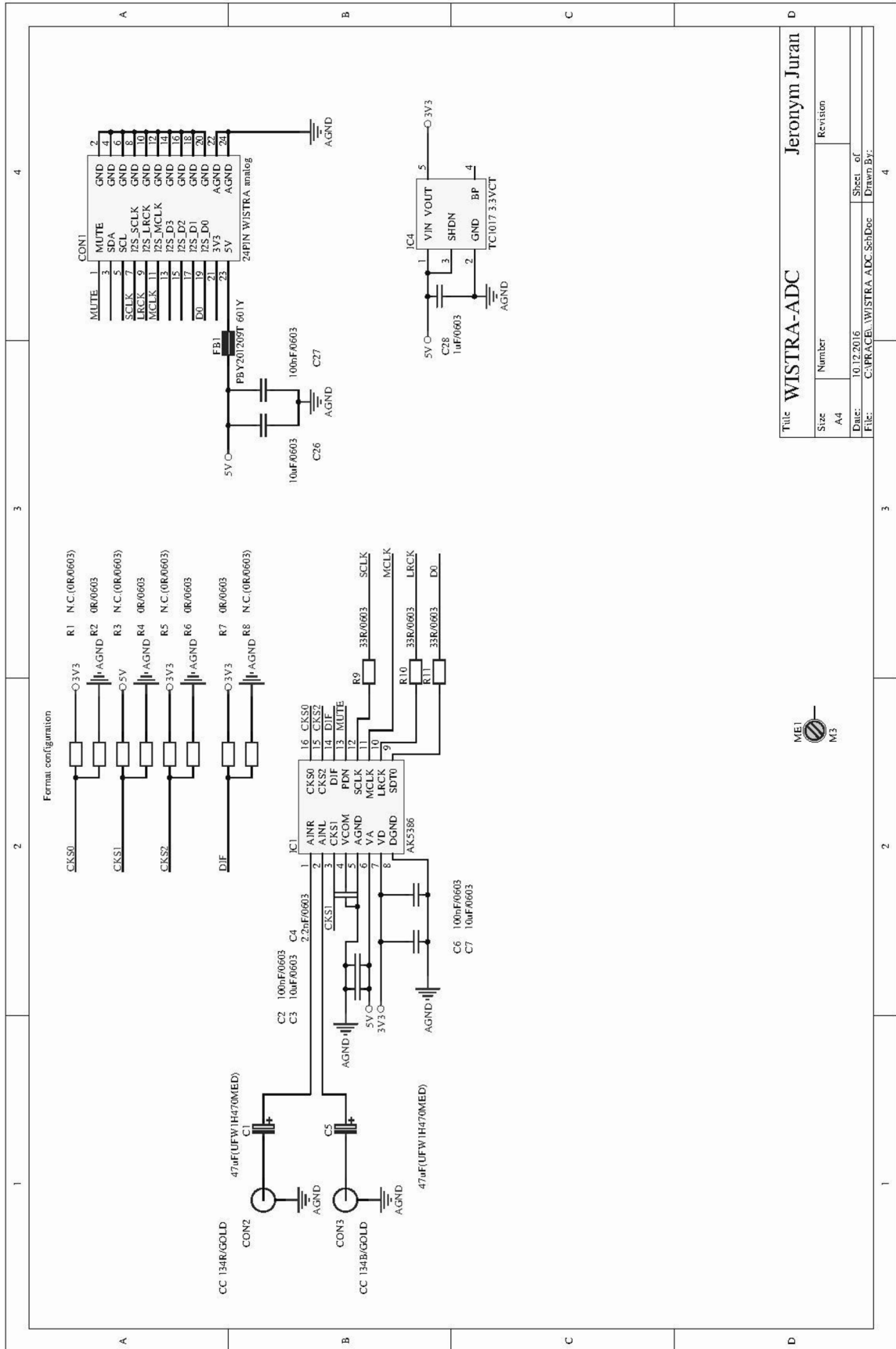
ADC	Analog to digital converter, analogově-digitální převodník
DAC	Digital to analog converter, digitálně-analogový převodník
DL	Display-list, seznam příkazů k zobrazení
DMA	Direct memory access, jednotka přímého přístupu do paměti
DPS	Deska plošných spojů.
EEPROM	Druh nevolatilní paměti paměti
EMC	Electromagnetic compatibility, elektromagnetická kompatibilita.
FIFO	First in,First out, paměť typu fronta.
FLASH	Druh nevolatilní paměti paměti
GPIO	General purpose input output.
GUI	Graphical user interface, grafické uživatelské rozhraní
I ² C	Inter-Integrated Circuit, sériové komunikační rozhraní
I ² S	Inter-IC Sound, rozhraní pro přenos audio signálu
MSB	Most significant bit, nejvýznamější bit
SPI	Serial Peripheral Interface, sériové komunikační rozhraní

A SCHÉMATA

A.1 Schéma vývojové desky vysílače



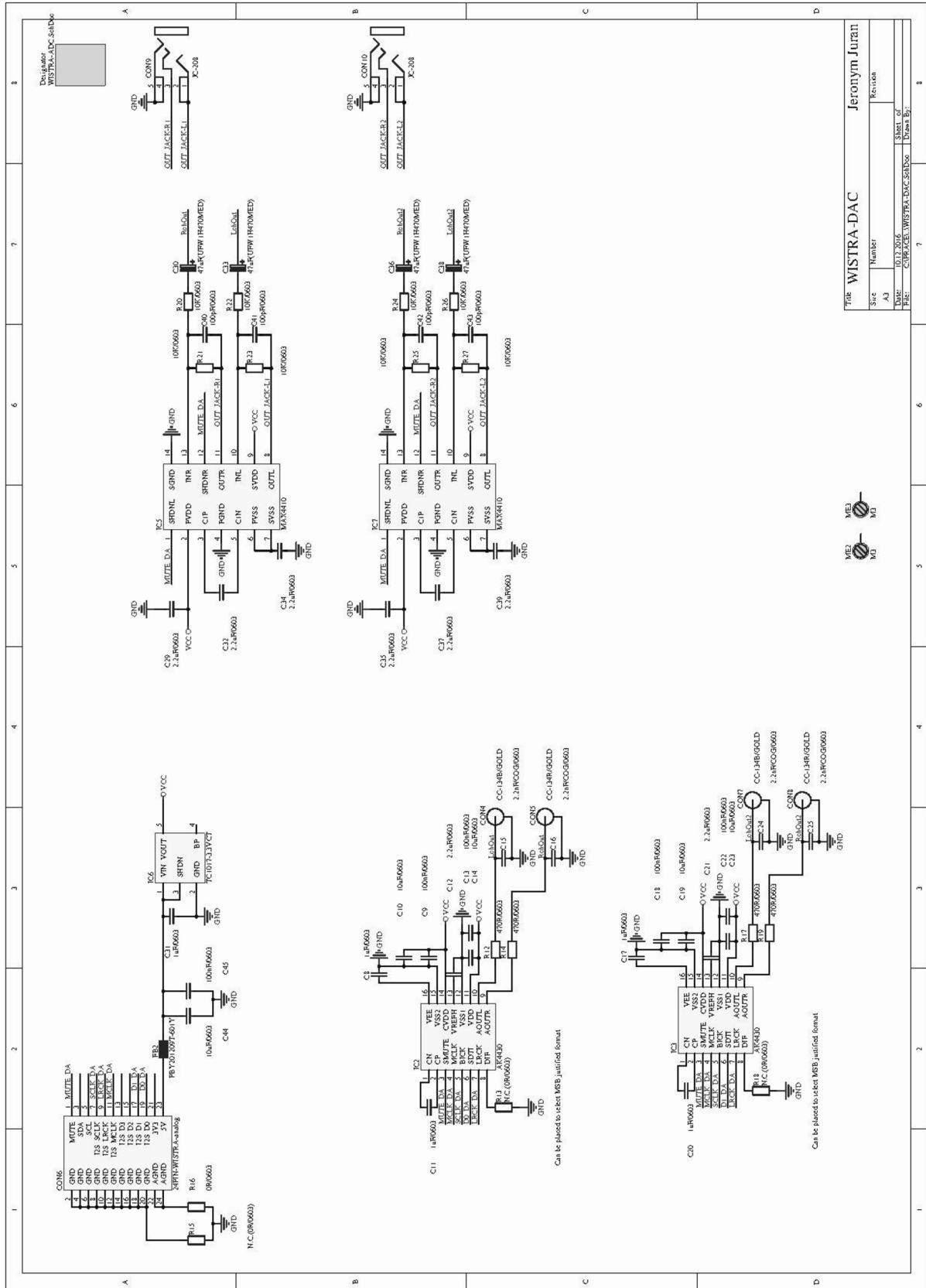
A.3 Schéma AD převodníku



Title		WISTRA-ADC	
Author		Jeronym Juran	
Size	Number	Revision	
A4			
Date:	01.12.2016	Sheet of	
File:	C:\PRACTE\WISTRA_ADC_SchDec	Drawn By:	



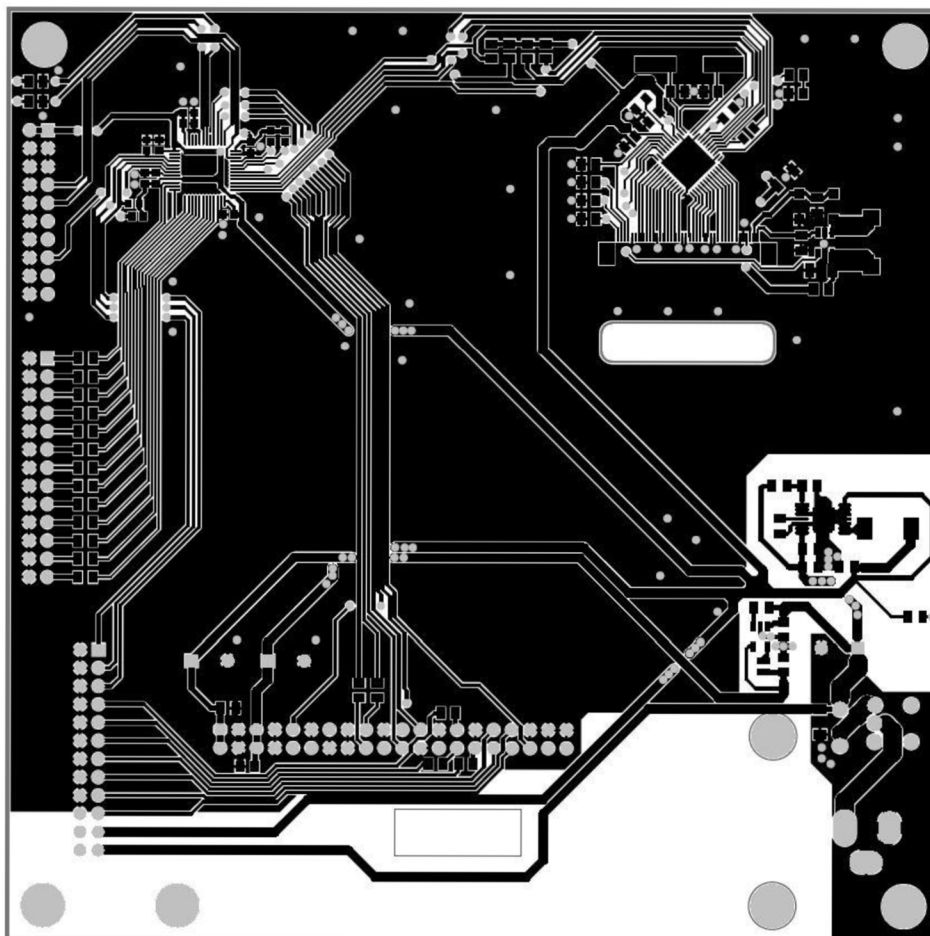
A.4 Schéma DA převodníku



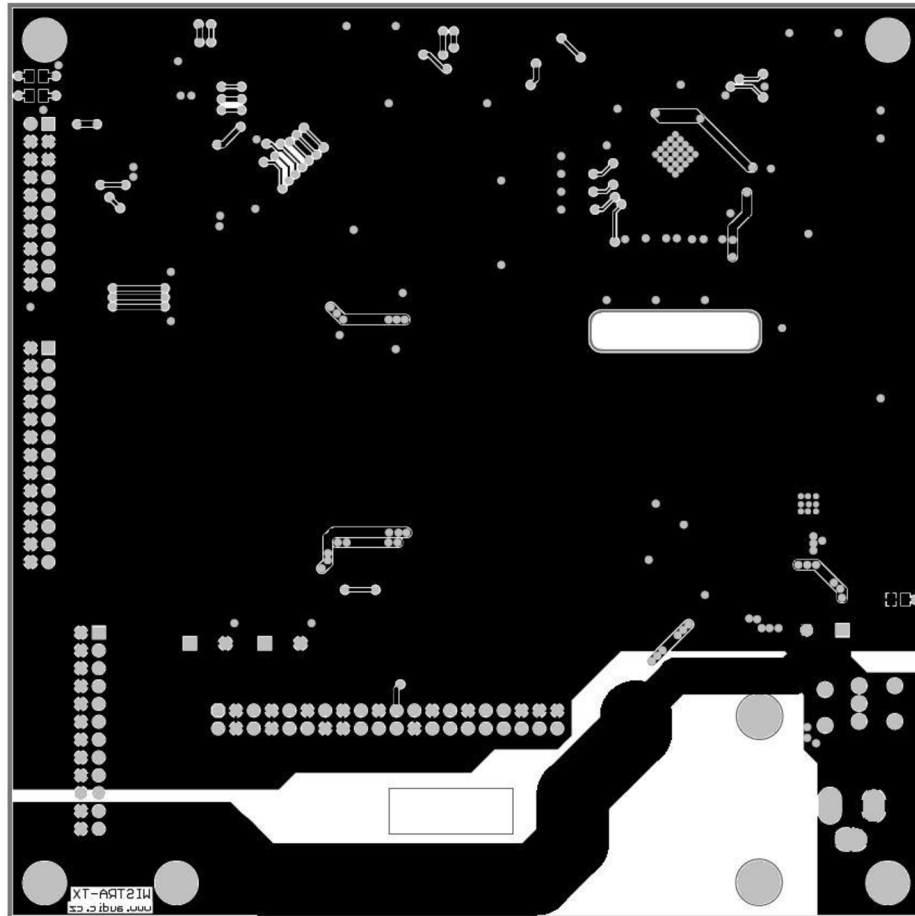
Title		WISTRA-DAC	
Author		Jeronym Juran	
Size	Number	Sheet of	Revision
A3		3	
Date	01.12.2016	Sheet of	
File:	C:\PRACTICE\WISTRA-DAC\SCH2	Drawn by:	

B ZOBRAZENÍ VODIVÝCH VRSTEV

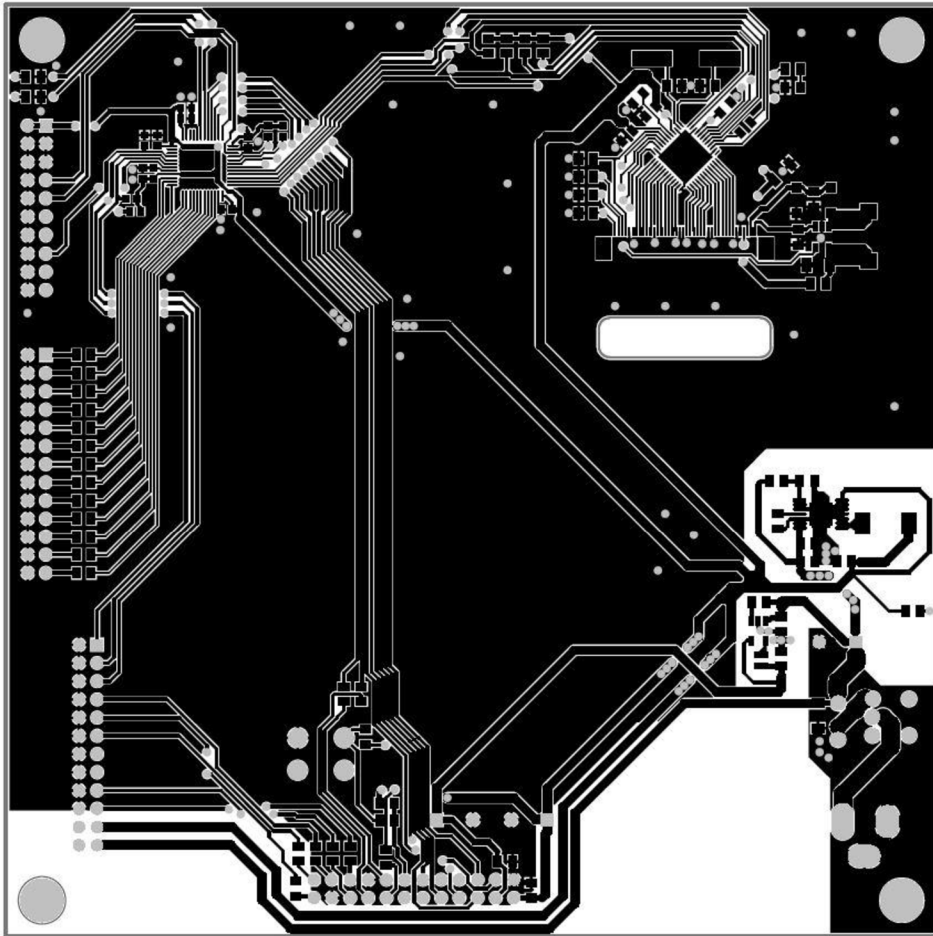
B.1 TX top



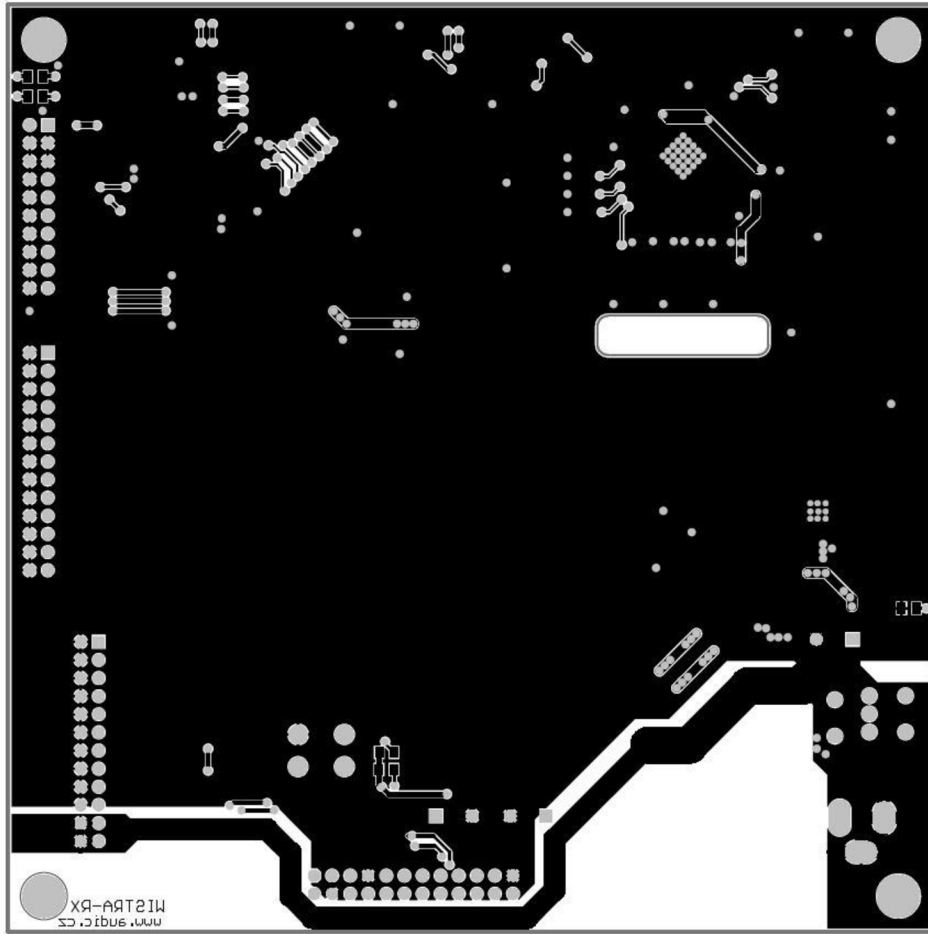
B.2 TX bot



B.3 RX top



B.4 RX bot



B.5 ADC/DAC

