

Modul USB FM rádia

USB FM Radio Modul

Tuto stránku nahradíte v tištěné verzi práce oficiálním zadáním Vaší diplomové či bakalářské práce.

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

V Ostravě 1. dubna 2016

.....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 1. dubna 2016

.....

Rád bych na tomto místě poděkoval zejména mému vedoucímu za bezbřehou trpělivost.

Abstrakt

Tato práce popisuje návrh USB FM přijímače se dvěma tunery. Jeden tuner slouží pro přehrávání zvuku a druhý pro vyhledávání dalších stanic. přijímač je v systému reprezentován jako USB zvuková karta.

Příjem je realizován dvojicí integrovaných obvodů Si4735-DU. Tyto jsou přes I²S a I²C spojeny s MCU PIC32MX250F128B, který přes USB zajišťuje komunikaci s počítačem. V rámci firmware MCU je, po neúspěchu s Microchip harmony frameworkem, napsán vlastní USB stack.

Knihovna je napsána v jazyku C s využitím knihovny libusb. Poskytuje funkce pro tři úrovně přístupu k tunerům.

Demonstrační aplikace je ve formě grafického uživatelského rozhraní, napsaná v C++ s využitím QT frameworku.

Vše je funkční pod OS Linux i Windows.

Klíčová slova: FM rádio, USB, RDS, QT, libusb, PIC

Abstract

This work describes design of USB FM radio receiver with two tuners. One tuner is for radio playback, second one seeks new stations. In computer, device acts as sound card. Receiving is done by couple of Si4735-DU integrated circuits, which are connected to MCU via I²C and I²S. MCU forwards data over USB to computer and back. Use of Microchip harmony framework was not successful so in firmware is USB stack written from scratch.

Library is written in C with use of libusb library. There are three levels of functions to access tuners.

Demo application has graphical user interface and is written in C++ in QT framework. All works under Linux and Windows.

Keywords: FM radio receiver, USB, RDS, QT, libusb, PIC

Seznam použitých zkratk a symbolů

AM	– Amlitudová Modulace (Rozhlasové vysílání v pásmu dlouhých vln)
CD	– Compact disc
DAB	– Digital Audio Broadcasting (Digitální pozemní rozhlasové vysílání)
DIP	– Dual Inline Package
FM	– Rozhlasové vysílání v pásmu velmi krátkých vln
I ² C	– Inter-Integrated Circuit
I ² S	– Integrated Interchip Sound
LW	– Long Waves (Rozhlasové vysílání v pásmu dlouhých vln)
MCU	– Microcontroller unit
PCM	– Pulse-code modulation
RDS	– Radio Data System
SPI	– TODO Serial Peripheral Interface ??
SSOP	– TODO
SW	– Short Waves (Rozhlasové vysílání v pásmu krátkých vln)
USB	– Universal Serial Bus
QFN	– TODO

Obsah

1	Úvod	5
2	Výběr součástek	6
2.1	Způsob příjmu rozhlasového vyslání	6
2.2	Volba rozhraní pro spojení modulu a počítače	6
2.3	Napojení tuneru na USB	7
2.4	Výsledná konstrukce	9
3	USB	10
3.1	Stručný úvod do full-speed USB 2.0	10
3.2	Microchip Harmony framework	13
3.3	Vlastní implementace	13
3.4	USB I ² C tunel	13
3.5	Omezení	13
4	Tuner	14
4.1	I2S	14
4.2	Ovládání tuneru	14
5	Knihovna	15
5.1	Nízko úrovněvé funkce	15
5.2	Středně úrovněvé funkce	15
5.3	Vysoko úrovněvé funkce	15
6	Závěr	16
7	Reference	17
	Přílohy	18
A	Schéma zapojení modulu	19

Seznam tabulek

1	Druhy USB přenosů.	10
---	----------------------------	----

Seznam obrázků

1	Blokové schéma TAS1020b. (Převzato z [13])	8
2	Blokové schéma zapojení.	9
3	USB endpointy a funkce.	11

Seznam výpisů zdrojového kódu

1	Vytvoření socketu.	12
---	----------------------------	----

1 Úvod

Tento text je ukázkou sazby diplomové práce v \LaTeX u pomocí třídy dokumentů `diploma`. Pochopitelně text není skutečnou diplomovou prací, ale jen ukázkou použití implementovaných maker v praxi. V kapitole ?? jsou ukázky použití různých maker a prostředí. V kapitole 6 bude „jako závěr“. Zároveň tato kapitola slouží jako ukázka generování křížových odkazů v \LaTeX u.

2 Výběr součástek

Vzhledem k tomu, že není možné se cenou zařízení přiblížit zavedeným výrobcům elektroniky, rozhodl jsme se výběr součástek a konstrukci modulu přizpůsobit tak, aby bylo možné modul vyrobit v domácích podmínkách.

2.1 Způsob příjmu rozhlasového vyslání

Jednou možností je řešení příjmu z diskretních součástek a nebo s pomocí analogových IO. Ovšem toto je příliš komplikované.

Na trhu je ovšem řada integrovaných obvodů, které zajišťují samotný příjem vysílání včetně vyhledávání static, měření kvality signálu a příjmu RDS a to s minimem potřebných externích součástek. Tyto IO se typicky ovládají pomocí I²C nebo SPI a zvuk poskytují digitálně přes rozhraní I²S a nebo analogově.

Bohužel drtivá většina je dostupná pouze v pouzdru QFN, které se velmi obtížně pájí a v minimální množství 1000 kusů. Výjimkou je SI4735-D60 od výrobce SILICON LABS, který je dostupný v pouzdru SSOP24 a je možné jej u nás zakoupit i po jednotlivých kusech. IO neumožňuje přijímat DAB, ale umí následující:

- Pásmo: FM, SW, MW, LW
- Vzorkovací frekvence až do 48kHz
- Rozlišení vzorku kanálu až do 24bitů
- Stereofonní příjem.
- Příjem RDS

2.2 Volba rozhraní pro spojení modulu a počítače

Po tomto rozhraní se budou přenášet dva druhy informací a to samotný zvuk a ovládání tunerů.

V současné době je prakticky jediným schůdným řešením použití rozhraní USB díky celé řadě výhod, které nabízí. Zejména jeho širokým rozšířením na téměř všech počítačích, od osobních přes servery až po jednodeskové či průmyslové počítače. Stejně tak je k dispozici velké množství součástek se zabudovanou podporou tohoto rozhraní. USB dále poskytuje možnost napájení připojených zařízení až do příkonu 2,5W. Má zabudovanou podporu pro různé druhy přenosů včetně isochronních (garantovaný periodický přenos předem dohodnutého množství dat). Specifikace USB zavádí standardní třídy funkcí v zařízení. V době psaní tohoto textu sice neexistuje třída pro ovládání tuneru, ale existuje třída popisující zvuková zařízení. Díky tomuto není potřeba vyvíjet vlastní ovladač zvukové karty na straně počítače.

2.2.1 Verze USB specifikací

V současné době je možné se setkat s USB verze 1.0, 1.1, 2.0 a 3.0. Dobrou zprávou je zpětná kompatibilita všech verzí. tj. zařízení podle specifikace 1.0 by mělo fungovat s jakýmkoliv hostem. Rychlost full speed definuje už první specifikace, její maximální propustnost 12Mbit je pro věrný přenos dvoukanálového zvuku více než dostatečná. Novější verze nepřinášejí žádnou vlastnost, která by byla pro tento projekt přínosná.

Odlíšná situace je v případě specifikací třídy USB audio. Existují vzájemně nekompatibilní verze 1.0 a 2.0. Ani zde mladší verze nepřináší žádný benefit, který bych mohl využít. Navíc doposud nemá nativní podporu ani ve Windows 10. Z tohoto důvodu není použití USB audio 2.0 příliš vhodné.

2.3 Napojení tuneru na USB

Požadavky:

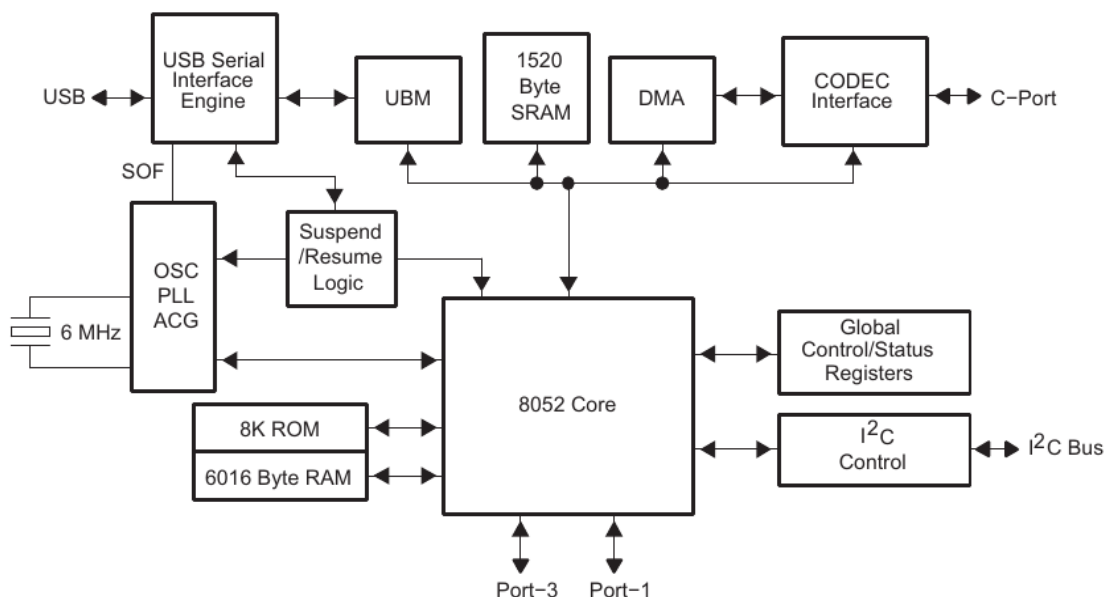
- Schopnost přenášet dvoukanálový zvuk beze znatelného zkreslení. Zvolil jsem PCM formát o vzorkovací frekvenci 48kHz a rozlišení 16bitů na jeden kanál. Pro srovnání audio CD používá 44,1kHz/16bitů.
- Alespoň jedno rozhraní I²S schopné přijímat zvuk a fungující v režimu master.
- Podporu pro USB audio. To implikuje nutnost podpory full speed USB a nebo rychlejší. Low speed nepodporuje isochronní přenosy, které jsou nezbytné pro přenos zvuku.
- Rozhraní I²C master pro ovládání tunerů.
- Kompatibilita s 3,3V logikou tunerů.

2.3.1 TAS1020b

[!htbp]

Jak je patrné z obrázku 1, jedná se o USB I²S zvukovou kartu a MCU v jednom. Na rozdíl od většiny MCU nemá interní paměť programu. Program se načítá při spuštění buď z eeprom paměti připojené přes I²C a nebo z přes USB ze zařízení ke kterému je obvod připojen.

Obvod podle specifikace [13] podporuje všechno potřebné. Full speed USB1.1 včetně USB audio 1.0, 14 endpointů z toho až dva mohou být isochronní. Dále až dvě vstupní I²S rozhraní a jednu I²C sběrnici. Nevýhodou je absence programové paměti, kusová dostupnost obvodu pouze ve formě vzorků a v mém případě také fakt, že s tímto druhem obvodů nemám žádné zkušenosti.



Obrázek 1: Blokové schéma TAS1020b. (Převzato z [13])

2.3.2 PIC16F1454

Je osmi bitový MCU od firmy Microchip s podporou full speed USB 2.0. Obvod obsahuje továrně kalibrovaný oscilátor a umí pracovat při napájecím napětí 2,3-5,5V. Díky tomu obvod nepotřebuje prakticky žádné externí součástky. V podstatě k němu stačí připojit pouze USB kabel.

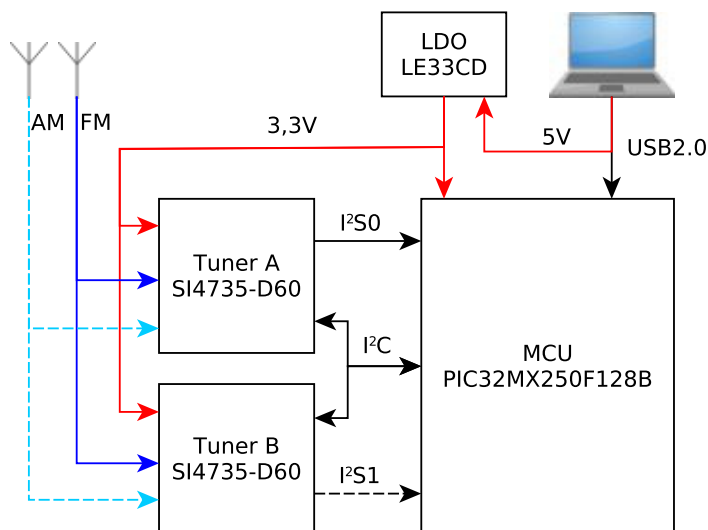
Výrobce poskytuje k tomuto MCU knihovnu Microchip Library for applications, které mimo jiné obsahuje implementaci USB audio 1.0. Navíc jedním ze vzorových projektů u této knihovny je i USB mikrofón, který řeší přenos zvuku do počítače.

Bohužel tento MCU nemá podporu I²S a na jeho softwarovou implementaci je příliš pomalý.

2.3.3 PIC32MX250F128

Tento 32 bitový MCU, taktéž od firmy Microchip, je vybaven všemi potřebnými rozhraními. Full speed USB 2.0, 2x nezávislé I²S, 2x I²C. Pracuje v rozmezí napájecích napětí 2,3-3,6V. V MCU jsou k dispozici čtyři DMA kanály, které je možné řetězit (po ukončení jednoho kanálu se automaticky spustí druhý). Podobně jako u PIC16F1454 je i k tomuto čipu je k dispozici framework Harmony [12] s podporou pro USB Audio 1.0. Je k dispozici v různých pouzdrech, dokonce i v DIP, které je možné přímo zapojit do nepájivého pole.

Pro modul jsem vybral právě tento MCU. Případně je možné něco málo ušetřit a použít PIC32MX220F032. Liší se pouze menšími velikostmi pamětí, konkrétně 32KB programové paměti místo 128KB a 8KB datové paměti namísto 32KB.



Obrázek 2: Blokové schéma zapojení.

2.4 Výsledná konstrukce

[!htbp]

Propojení jednotlivých komponent je naznačeno na obrázku 2. Jak je patrné, zahrnul jsem i propojení tuneru B s MCU přes I²S a také vyvedení anténních AM vstupů tunerů (v obrázku čárkovaně). Aktuálně nejsou využity, ale v budoucnu bude možné zařízení rozšířit o podporu příjmu všech pásem, které tunery podporují a nebo přidat režim, kdy se modul bude chovat jako dvě nezávislé zvukové karty.

Celé zařízení je napájeno z USB přes jediný lineární stabilizátor LE33CD, který snižuje napájecí napětí na 3,3V. Dokáže poskytnout až 100mA a je odolný proti nadproudu a přehřátí [14]. MCU je zapojen podle doporučení v katalogovém listu [6] kapitola 2.1. Stejně tak tunery jsem zapojeny podle doporučení v katalogovém listu [4] kapitola 2.2.

Tady bych chtěl zmínit vstup RCLK. Tuner potřebuje v době ladění hodinový signál. V katalogovém listu je několikrát zmíněno, že tento signál je nutné buď přivést přímo na vstup RCLK a nebo se získá zapojením 32,768kHz krystalu. Až v programovací příručce [5] v popisu property `REFCLK_PRESCALE` je možnost odvodit hodinový signál od vstupu hodinového signálu I²S DCLK. Této možnosti jsem samozřejmě využil.

Kompletní schéma zapojení je součástí příloh.

3 USB

Na rozdíl například od například populárního rozhraní UART je USB podstatně komplikovanější. Je to určitou daní za jeho univerzálnost. Vzhledem k velkému rozsahu specifikace USB [?] se omezím pouze na popis částí nezbytných pro implementaci modulu.

S vydáním specifikace USB 2.0 byly předchozí specifikace označeny jako zastaralé a neměly by se používat pro nové konstrukce. Následující text se tedy týká USB 2.0 a rychlosti full-speed.

3.1 Stručný úvod do full-speed USB 2.0

3.1.1 Topologie

[!htbp]

Ačkoliv v název rozhraní (Univerzální Sériová Sběrnice) napovídá, že jde o sběrnici, jedná se o zapojení typu hvězda. Přesněji, jak je vidět z obrázku ??, připojená zařízení a rozbočovače tvoří strom jehož kořenem je hostitel. Tento implementuje tzv. kořenový rozbočovač (root hub), ke kterému je možné buď přímo připojit jedno zařízení a nebo rozbočovač a do něj dalších až osm zařízení/rozbočovačů. je možné takto za sebe zřetězovat až pět rozbočovačů. Celkově je možné na jeden kořenový rozbočovač připojit až 127 zařízení (včetně rozbočovačů).

Velkou výhodou je, že zařízení je od topologie odstíněno. Vždy se z jeho pohledu komunikuje přímo s hostem.

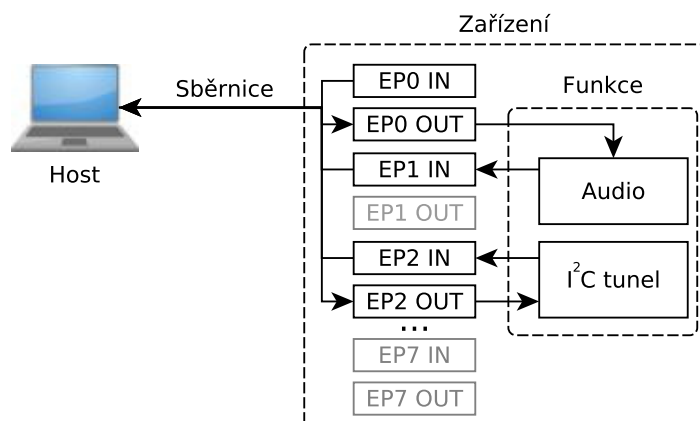
Ve specifikaci je také zohledněn fakt, že zařízení zpravidla nezastává pouze jedinou funkci. To je konec konců případ i tohoto modulu. Obsahuje dvě funkce, zvukovou kartu a I²C tunel pro komunikaci s tunery.

3.1.2 Komunikace

	Latence	Vyhrazená šířka pásma	Spolehlivý přenos	Typ dat
Isochronní přenos	Minimální	až 90%	Ne	Proud
Hromadný přenos	Negarantovaná	Ne	Ano	Proud
Přenos přerušení	Minimální	až 90%	Ano	Proud
Řídící přenos	Negarantovaná	až 10%	Ano	Zprávy

Tabulka 1: Druhy USB přenosů.

Aby bylo možné uspokojit nároky na přenos (transfer) dat rozdílné povahy různými funkcemi, zavádí specifikace koncové body (endpoint). Osm výstupních (OUT) pro směr z hostitele do zařízení a osm pro směr opačný (IN). Směr je vždy určován právě z pohledu hostitele. Každému bodu je možné přiřadit jeden ze čtyř druhů přenosu podle tabulky



Obrázek 3: USB endpointy a funkce.

3.1.2. Výjimkou je Vstupní a výstupní koncový bod nula. Tyto vždy slouží pro řídicí přenosy a na rozdíl od ostatních bodů je musí podporovat všechna zařízení. Full-speed USB podporuje čtyři druhy komunikace uvedené v tabulce 3.1.2.

Body nula jsou využívány jednak k inicializaci a správě vlastního zařízení, ale také mohou být využívány zároveň i funkcemi. Například popisovaný modul jej využívá pro ovládání audio funkce. Modul dále využívá vstupní koncový bod 1 pro přenos zvuku do hostitele a potom dvojici koncových bodů 2 v obou směrech pro komunikaci s tunery viz. obrázek 3.

Přenos vždy sestává z alespoň jedné transakce (transaction), která se dále dělí na pakety (packets). Transakce je z pohledu zařízení vždy vyřizována od počátku do konce bez přerušení jinou transakcí. Je vždy iniciována vysláním token paketu hostitelem, který takto může řídit šířku pásma přidělovanou jednotlivým zařízením na sběrnici. Token pakety mohou být podle druhu transakce následujícími třemi druhy:

- **IN** (Vstupní) - Následuje přenos ze zařízení do hostitele.
- **OUT** (Výstupní) - Následuje přenos z hostitele do zařízení.
- **SETUP** - Následuje řídicí přenos.

Za tímto paketem následuje nula nebo více paketů s daty. Aby bylo možné detekovat výpadek nebo duplikaci některého z paketů jsou specifikovány hned dva typy paketů. A to DATA0 a DATA1. U izochronních transakcí se vždy posílají pakety DATA0. U ostatních transakcí se vyše nejprve DATA0 a poté se tyto druhy paketů střídají nezávisle na transakcích. Znovu se od DATA0 začne pouze v následujících případech:

- Na začátku každé řídicí transakce.
- Po následujících žádostech hostitele (bude popsáno dále):

- Přiřazení konfigurace.
- Zrušení zastavení koncového bodu.
- Nastavení rozhraní.

Za datovými pakety následuje potvrzování transakce protistranou. Izochronní přenosy potvrzování nepodporují, tudíž datovými pakety jejich transakce končí. Transakce hromadných přenosů a přenosu přerušeni jsou vždy zakončeny jedním potvrzovacím (handshake) paktem. Transakce řídicích přenosů mají před potvrzovací paket vložen jeden datový paket nulové délky (zero length packet často zkracovaný ZLP), ale opačného směru než všechny předchozí datové pakety. Potvrzovací paket vždy vysílá zařízení. Má vždy jeden z následujících typů:

- **ACK** (Úspěch) - Úspěšné ukončení transakce.
- **NAK** (Neúspěch) - Typicky poškozená přijímaná data a nebo častěji zařízení nemá připravena data k odeslání.
- **STALL**(Chyba) - Zařízení takto reaguje na požadavek, který nepodporuje.

Rozdíl mezi NAK a STALL je, že po NAK potvrzení bude hostitel požadavek opakovat (počet opakování není explicitně specifikován), kdežto STALL signalizuje nemožnost vyřízení požadavku a tudíž jej opakovat nemá smysl.

Formátování a rozpoznávání paketů řeší přímo USB modul v Mikrokontroléru, není tedy nutné se jím hlouběji zabývat. Detailní popis je v kapitole 8 USB specifikace [2].

3.1.3 Dekriptory

USB specifikace zavádí deskriptory (descriptors). Jedná se o unifikovaný způsob jak může zařízení informovat hostitele o svých schopnostech a požadavcích. Na základě právě těchto informací může operační systém vybrat pro funkce zařízení odpovídající ovladače, řadič v hostiteli se dozví kolik dat, jak často bude přenášet po jednotlivých koncových bodech a jakou má tomuto přenosu přiřadit prioritu a podobně.

Deskriptory jsou binární struktury takže je nutné vzít v potaz endianitu. U USB je to vždy little-endian, to znamená, že se čílo odesílá od nejméně významného bytu po nejvíce významný. Například 0x1234 se přenese nejdříve 0x34 a poté 0x12. Ačkoliv MCU PIC32 s použitím dodávaného kompilátoru XC32 používá taktéž little-endian, je z hlediska případné přenositelnosti výhodné implementovat jako pole typu **char** viz ukázka ??.

```
const static Usb_descriptor_device usb_desc_device =
{
    .bLength           = sizeof(Usb_descriptor_device),
    .bDescriptorType    = USB_DSC_DEVICE,
    .bcdUSB             = 0x0200, //2.0
    .bDeviceClass       = 0x00,
    .bDeviceSubClass    = 0x00,
    .bDeviceProtocol    = 0x00,
```

```
.bMaxPacketSize0 = 64,  
.idVendor         = USB_VID,  
.idProduct        = USB_PID,  
.bcdDevice        = 0x0003,  
.iManufacturer    = 1,  
.iProduct         = 2,  
.iSerialNumber    = 0,  
.bNumConfigurations = 1  
};
```

Výpis 1: Deskriptor zařízení.

3.1.4 USB audio 1.0

3.2 Microchip Harmony framework

Nepoužitelnost Harmony frameworku

3.3 Vlastní implementace

3.4 USB I²C tunel

i2c -> usb

3.5 Omezení

4 Tuner

4.1 I2S

Popis

Problém synchronizace hodin

4.2 Ovládání tuneru

4.2.1 RDS

čtení z tuneru

dekódování základních informací

5 Knihovna

5.1 Nízko úrovněvé funkce

5.2 Středně úrovněvé funkce

5.3 Vysoko úrovněvé funkce

5.3.1 RDS dekodér

6 Závěr

Tak doufám, že Vám tato ukázka k něčemu byla. Další informace najdete v publikacích

Bc. Pavel Kovář

7 Reference

- [1] AXELSON, Jan. *USB complete: the developer's guide*. 4th ed. Madison, WI: Lakeview Research, 2009, xxiii, 504 p. ISBN 1-931448-08-6.
- [2] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips. *Universal Serial Bus Specification: Revision 2.0* [online] 2000-04-27 [2015-12-26] http://www.usb.org/developers/docs/usb20_docs/usb_20_0702115.zip
- [3] Gal Ashour, Billy Brackenridge, Oren Tirosh, Altec Lansing, Craig Todd, Remy Zimmermann, Geert Knapen. *Universal Serial Bus Device Class Definition for Audio Devices: Release 1.0* [online] 1998-03-18 [2015-12-26] http://www.usb.org/developers/docs/devclass_docs/audio10.pdf
- [4] Silicon Laboratories, *Si4730/Si4731/Si4734/Si4735-D60 Broadcast AM/FM/SW/LW Radio Receiver: Rev. 1.2 8/13* [online] 2013-08-08 [2015-12-26] <https://www.silabs.com/Support%20Documents/TechnicalDocs/Si4730-31-34-35-D60.pdf>
- [5] Silicon Laboratories, *AN332: Si47xx Programming Guide: Rev. 1.0 9/14* [online] 2014-09-10 [2015-12-26] <http://www.silabs.com/Support%20Documents/TechnicalDocs/AN332.pdf>
- [6] Microchip Technology Inc. *PIC32MX1XX/2XX Family Data Sheet: Revision H* [online] 2015-07-29 [2015-12-26] <http://ww1.microchip.com/downloads/en/DeviceDoc/60001168H.pdf>
- [7] Microchip Technology Inc. *PIC32 Family Reference Manual, Sect. 23 Serial Peripheral Interface* [online] 2011-10-11 [2015-12-26] <http://ww1.microchip.com/downloads/en/DeviceDoc/61106G.pdf>
- [8] Microchip Technology Inc. *PIC32 Family Reference Manual, Sect. 24. Inter-Integrated Circuit* [online] 2013-03 [2015-12-26] <http://ww1.microchip.com/downloads/en/DeviceDoc/61116F.pdf>
- [9] Microchip Technology Inc. *PIC32 Family Reference Manual, Sect. 31 DMA Controller* [online] 2013-11-15 [2015-12-26] <http://ww1.microchip.com/downloads/en/DeviceDoc/60001117H.pdf>
- [10] Microchip Technology Inc. *PIC32 Family Reference Manual, Sect. 27 USB On-The-Go* [online] 2011-04-13 [2015-12-26] <http://ww1.microchip.com/downloads/en/DeviceDoc/61126F.pdf>
- [11] Microchip Technology Inc. *PIC32MX1XX/2XX 28/36/44-pin Family Silicon Errata and Data Sheet Clarification* [online] 2015-07-29 [2015-12-26] <http://ww1.microchip.com/downloads/en/DeviceDoc/80000531G.pdf>

- [12] Microchip Technology Inc. *MPLAB Harmony USB Libraries Help* [online] 2012-11-15 [2015-12-26] <http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB%20Harmony%20USB%20Libraries%20%28v1.06.02%29.pdf>
- [13] *TAS1020B USB Streaming Controller* [online] 2011-05 [2015-12-29] <http://www.ti.com/lit/ds/symlink/tas1020b.pdf>
- [14] STMicroelectronics *Very low-dropout voltage regulator with inhibit function* [online] 03-2014 [2016-01-09] <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000545.pdf>

A Schéma zapojení modulu

Na štorc