

# **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 I<sup>2</sup>S a I<sup>2</sup>C 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 I<sup>2</sup>C and I<sup>2</sup>S. 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 <sup>2</sup> C	– Inter-Integrated Circuit
I <sup>2</sup> 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
UTF-16	– Způsob kódování znaků ISO 10646/Unicode
QFN	– TODO

## Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Výběr součástek</b>	<b>6</b>
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
<b>3</b>	<b>USB</b>	<b>10</b>
3.1	Stručný úvod do full-speed USB 2.0 . . . . .	10
3.2	Microchip Harmony framework . . . . .	22
3.3	Vlastní implementace . . . . .	22
3.4	USB I <sup>2</sup> C tunel . . . . .	22
3.5	Omezení . . . . .	23
<b>4</b>	<b>Tuner</b>	<b>24</b>
4.1	I2S . . . . .	24
4.2	Ovládání tuneru . . . . .	24
<b>5</b>	<b>Knihovna</b>	<b>25</b>
5.1	Nízko úrovněvé funkce . . . . .	25
5.2	Středně úrovněvé funkce . . . . .	25
5.3	Vysoko úrovněvé funkce . . . . .	25
<b>6</b>	<b>Závěr</b>	<b>26</b>
<b>7</b>	<b>Reference</b>	<b>27</b>
	<b>Přílohy</b>	<b>28</b>
<b>A</b>	<b>Schéma zapojení modulu</b>	<b>29</b>

## Seznam tabulek

1	Druhy USB přenosů. . . . .	10
2	Deskriptor zařízení. . . . .	13
3	Deskriptor konfigurace. . . . .	15
4	Deskriptor řídicího rozhraní zvuku. . . . .	16
5	Deskriptor řídicího rozhraní zvuku - hlavička. . . . .	16
6	Deskriptor řídicího rozhraní zvuku - vstupní terminál. . . . .	17
7	Deskriptor řídicího rozhraní zvuku - výstupní terminál. . . . .	17
8	Deskriptor rozhraní odesílání zvuku. . . . .	18
9	Deskriptor rozhraní odesílání zvuku. . . . .	18
10	Deskriptor formátu zvuku - hlavička. . . . .	19
11	Deskriptor formátu zvuku. . . . .	19
12	Deskriptor koncového bodu odesílání zvuku. . . . .	20
13	Deskriptor koncového bodu odesílání zvuku specifický pro danou třídu. .	21



## Seznam obrázků

1	Blokové schéma TAS1020b. (Převzato z [15]) . . . . .	8
2	Blokové schéma zapojení. . . . .	9
3	USB endpointy a funkce. . . . .	11

## Seznam výpisů zdrojového kódu

1	Deskriptor zařízení. . . . .	22
---	------------------------------	----

## 1 Úvod

Tento text je ukázkou sazby diplomové práce v  $\text{\LaTeX}$ 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  $\text{\LaTeX}$ 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 vysílání

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

Na trhu je 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 I<sup>2</sup>C nebo SPI a zvuk poskytují digitálně přes rozhraní I<sup>2</sup>S 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.

Odlišná 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 I<sup>2</sup>S 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 I<sup>2</sup>C master pro ovládání tunerů.
- Kompatibilita s 3,3V logikou tunerů.

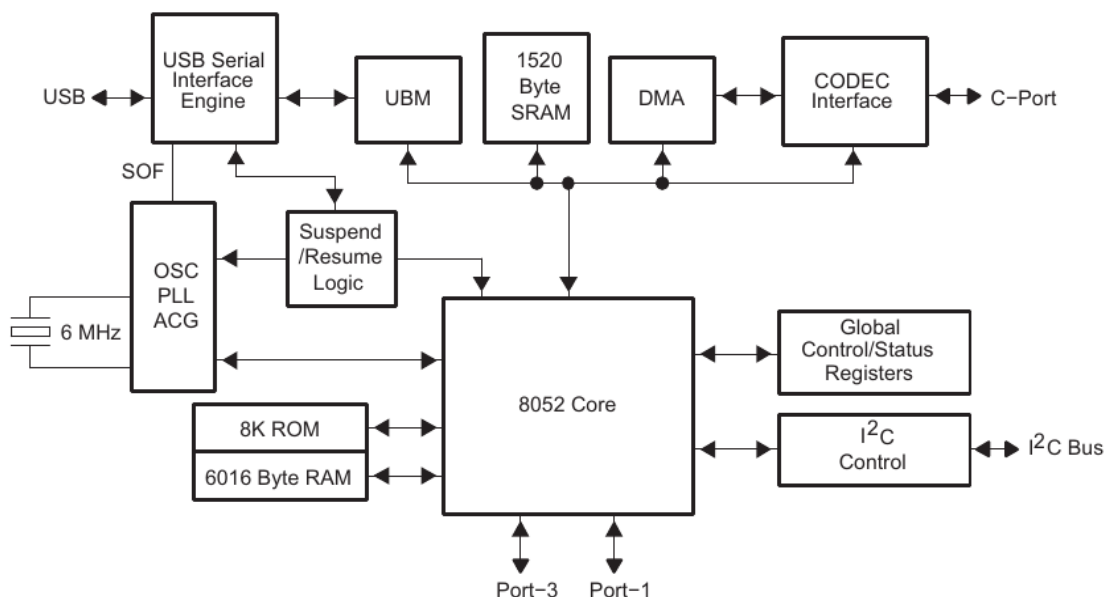
### 2.3.1 TAS1020b

Jak je patrné z obrázku 1, jedná se o USB I<sup>2</sup>S zvukovou kartu a MCU v jednom. Na rozdíl od většiny MCU nemá interní paměť programu. Program se načítá při spuštění buď z E<sup>2</sup>PROMpaměti připojené přes I<sup>2</sup>C a nebo přes USB ze zařízení, ke kterému je obvod připojen.

Obvod podle specifikace [15] podporuje všechno potřebné. Full speed USB1.1 včetně USB audio 1.0, 14 endpointů z toho až dva mohou být isochronní. Dále nabízí až dvě vstupní I<sup>2</sup>S rozhraní a jednu I<sup>2</sup>C 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.

### 2.3.2 PIC16F1454

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.



Obrázek 1: Blokové schéma TAS1020b. (Převzato z [15])

Díky tomu obvod nepotřebuje prakticky žádné externí součástky. V podstatě k němu stačí připojit pouze USB kabel.

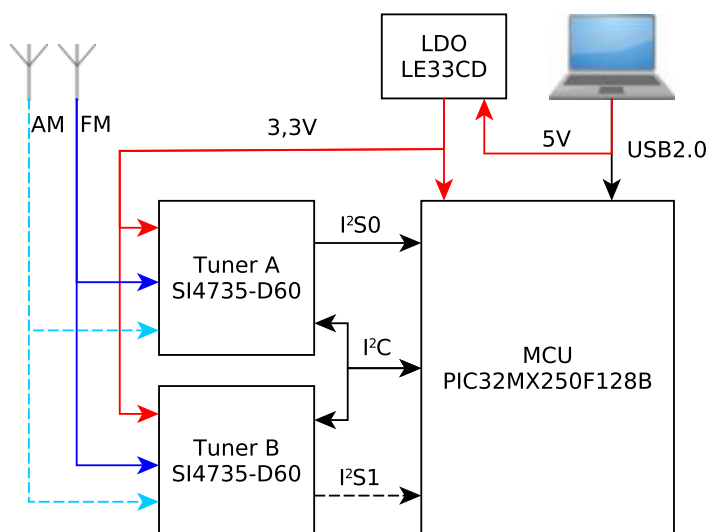
Výrobce poskytuje k tomuto MCU knihovnu Microchip Library for applications, která mimo jiné obsahuje implementaci USB audio 1.0. Navíc jedním ze vzorových projektů u této knihovny je i USB mikrofón, který řeší přenos zvuku do počítače.

Bohužel tento MCU nemá podporu I<sup>2</sup>S 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 I<sup>2</sup>S, 2x I<sup>2</sup>C. 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 k dispozici framework Harmony [14] s podporou pro USB Audio 1.0. Vyrábí se 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

Propojení jednotlivých komponent je naznačeno na obrázku 2. Jak je patrné, zahrnul jsem i propojení tuneru B s MCU přes I<sup>2</sup>S 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 [16]. MCU je zapojen podle doporučení v katalogovém listu [8] kapitola 2.1. Stejně tak tunery jsem zapojeny podle doporučení v katalogovém listu [6] 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 [7] v popisu property `REFCLK_PRESCALE` je možnost odvodit hodinový signál od vstupu hodinového signálu I<sup>2</sup>S DCLK. Této možnosti jsem samozřejmě využil.

Kompletní schéma zapojení je součástí příloh.

## 3 USB

Na rozdíl od například populárního rozhraní UART je USB podstatně komplikovanější. Je to určitou daní za jeho univerzálnost. Vzhledem k velkému rozsahu specifikace USB [2] 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

TODO Obrázek s pyramidou topologie

Ačkoliv 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 I<sup>2</sup>C tunel pro komunikaci s tunery.

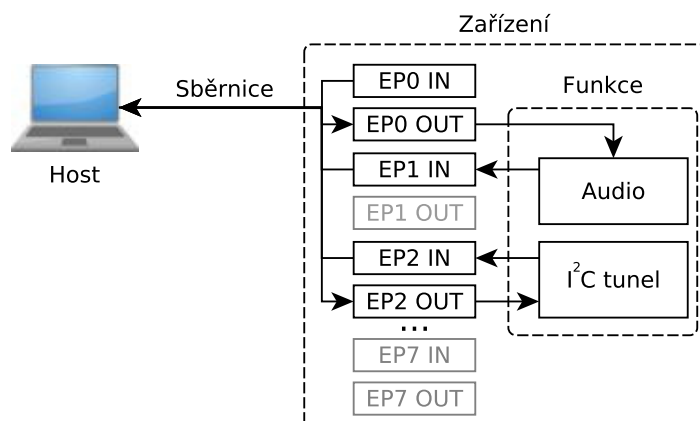
#### 3.1.2 Komunikace

	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 1. Výjimkou je vstupní a výstupní koncový bod nula. Tyto vždy slouží pro řídicí





Obrázek 3: USB endpointy a funkce.

přenosy a na rozdíl od ostatních bodů je musí podporovat všechna zařízení. Full-speed USB podporuje čtyři druhy komunikace uvedené v tabulce 1.

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 tokenu paketu hostitelem, který takto může řídit šířku pásma přidělovanou jednotlivým zařízením na sběrnici. Token pakety mohou být podle druhu transakce následujících třech druhů:

- **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 přerušení jsou vždy zakončeny jedním potvrzovacím (handshake) paketem. 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 Deskriptory

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.

Při vývoji zařízení je základním rozdělením deskriptorů rozdělení podle požadavků hostitele:

- **Deskriptor zařízení** (Service descriptor) - Nejnutnější informace pro správu zařízení na sběrnici.
- **Deskriptory řetězců** (String descriptors) - Pole textových řetěců a informace o dostupných lokalizacích.
- **Deskriptory konfigurace** (Configuration descriptor) - Struktura deskriptorů s veškerými dalšími informacemi.

Popis všech deskriptorů všech možných funkcí je zcela mimo rozsah tohoto textu. Dále se omezím pouze na deskriptory a jejich hodnoty použité v modulu.

Název pole	Délka	Hodnota	
bLength	1B	18	Délka deskriptoru.
bDescriptorType	1B	0x01	Typ deskriptoru.
bcdUSB	2B	0x0200	Verze USB specifikace implementovaná zařízením. (2.0)
bDeviceClass	1B	0x00	Třída zařízení. 0x00 znamená, že třídu specifikuje každé rozhraní zvlášť.
bDeviceSubClass	1B	0x00	Podtřída zařízení. Pokud je bDeviceClass 0x00, musí být i toto pole 0x00.
bDeviceProtocol	1B	0x00	Protokol zařízení. Pokud je bDeviceClass 0x00, musí být i toto pole 0x00.
bMaxPacketSize	1B	64	Největší délka data, kterou je možné odeslat koncovým bodem 0.
idVendor	2B	0x04D8	ID Výrobce zařízení.
idProduct	2B	0xF32C	ID zařízení.
bcdDevice	2B	0x0100	Verze zařízení 1.0.
iManufacturer	1B	1	Odkaz na řetězec s názvem výrobce.
iProduct	1B	2	Odkaz na řetězec s názvem zařízení.
iSerialNumber	1B	0	Odkaz na řetězec se sériovým číslem zařízení. 0 znamená nespecifikován.
bNumConfigurations	1B	1	Počet konfigurací zařízení.

Tabulka 2: Deskriptor zařízení.

### 3.1.4 Deskriptor zařízení

V tabulce 2 je uveden deskriptor zařízení tak, jak je použit v modulu. Myslím, že popis významu polí v tabulce je dostatečný. Za zmínku stojí ID výrobce a zařízení. Jejich účel je stejný jak například MAC adresa síťových zařízení a to jednoznačně identifikovat druh zařízení. Oficiální cesta je požádat o přiřazení ID výrobce USB implementers fórum, což v době psaní toho textu stojí 5000 amerických dolarů [17]. Výrobci programovatelných součástek s podporou USB, ale naštěstí z pravidla nabízejí možnost zdarma získat ID produktu z jejich rozsahů. Jednou z podmínek bývá nutnost použít přidělení ID právě na jejich součástce. Pro modul jsem získal ID produktu od firmy Microchip 0xF32C. ID výrobce je 0x04D8.

Pole iManufacturer, iProduct a iSerialNumber nesou indexy na deskriptory textových řetězců. Všechna pole jsou nepovinná. V případě jejich vynechání se použije index s hodnotou 0. Jak napovídají názvy jednotlivých polí, je možné takto přidat popis výrobce a zařízení ve formě lidsky čitelného textu a sériové číslo daného kusu zařízení, které může využít ovladač v hostiteli například pro načtení posledního nastavení po opětovném připojení zařízení.

### 3.1.5 Deskriptory řetězců

TODO: Tabulka se zkrácenými string deskriptory.

Deskriptory řetězců jsou organizovány jako pole indexované od nuly. Každý jeden řetězec začíná hlavičkou deskriptoru ve které je určen typ deskriptoru (0x??) a jeho celková délka v bytech. Poté následuje samotný text zakódovaný podle normy unicode, konkrétně UTF-16. Je tedy možné použití i národních znaků.

Jedinou výjimkou je deskriptor s indexem 0. V zařízení je možné mít více sad textů v různých jazycích. Seznam dostupných jazyků (jazykových kódů) je právě v tomto deskriptoru. Seznam těchto kódů popisuje [5]. V modulu jsem se omezil pouze na angličtinu (kód 0x0409).

### 3.1.6 Konfigurace zařízení

Konfigurace zařízení je soubor deskriptorů který popisuje schopnosti zařízení a také jeho nároky na přenos popřípadě definují sady parametrů z nichž si může hostitel vybrat tu, která mu v daný okamžik nejvíce vyhovuje. Zařízení musí specifikovat minimálně jednu konfiguraci, ale také více. Hostitel potom zařízení jednu přidělí případně ji může kdykoliv změnit za jinou.

Každá jedna konfigurace začíná Deskriptorem konfigurace za kterým následují deskriptory rozhraní a koncových bodů do kterých mohou být zanořeny deskriptory další. Tvoří takto stromovou strukturu. V případě mého modulu je možná pouze jediná konfigurace. Její struktura vypadá následovně:

1. Deskriptor řídicího rozhraní zvuku.
  - (a) Deskriptor řídicího rozhraní zvuku - hlavička.
  - (b) Deskriptor řídicího rozhraní zvuku - vstupní terminál přijímač rádia.
  - (c) Deskriptor řídicího rozhraní zvuku - výstupní terminál odesílání zvuku přes USB.
2. Deskriptor rozhraní pro odesílání zvuku - varianta s vypnutým přenosem.
3. Deskriptor rozhraní pro odesílání zvuku - varianta se zapnutým přenosem.
  - (a) Deskriptor rozhraní pro odesílání zvuku - obecný deskriptor.
  - (b) Deskriptor rozhraní pro odesílání zvuku - popis formátování dat.
  - (c) Deskriptor koncového bodu - odesílání zvuku.
4. Deskriptor rozhraní specifikovaného výrobcem - I<sup>2</sup>C tunel.
  - (a) Deskriptor koncového bodu - odesílání dat hostu.
  - (b) Deskriptor koncového bodu - příjem dat z hosta.

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x02	Typ deskriptoru.
wTotalLength	2B	127	Celková délka všech deskriptorů konfigurace.
bNumInterfaces	1B	3	Počet rozhraní v konfiguraci.
bConfigurationValue	1B	1	Index této konfigurace.
iConfiguration	1B	0	Odkaz na řetězec s popisem konfigurace.
bmAttributes	1B	0b10000000	Bitová maska s atributy.

Tabulka 3: Deskriptor konfigurace.

### 3.1.7 Deskriptor konfigurace

Jak je vidět z tabulky 3, tento deskriptor slouží zejména pro informace pro identifikaci konfigurace. A to zejména index `iConfiguration`, to je hodnota, kterou poté pošle host do zařízení v požadavku o přidělení konfigurace. Dále pak `wTotalLength`. Hostitel obdrží všechny případné konfigurace v jednom bloku. Na základě této hodnoty rozliší kde jednotlivé konfigurace začínají a končí.

Za zmínku také stojí pole `bmAttributes`. Nejvyšší bit musí být z důvodu kompatibility s USB 1.0 nastaven na 1, nejnižší bity 0-4, jsou rezervovány pro budoucí použití a musí být nastaveny na 0. Bit 6 signalizuje, že zařízení není napájeno z USB sběrnice. Bit 5 signalizuje že zařízení chce využívat mechanismus vlastního probuzení a informování hostitele o události. Modul má oba atributy nastaveny na 0.

### 3.1.8 USB audio 1.0

Největší část konfigurace zabírají deskriptory popisující část přenosu zvuku. Je to dáno také tím, že specifikace USB audio [3] nepopisuje pouze přenos audia po USB. Nabízí také prostředky pro popis topologie vstupů, výstupů, různých efektových jednotek přepínačů, směšovačů a podobně včetně jejich ovládání. Bylo by například možné takto realizovat kompletní ovládání mixážního pultu kde by přes usb mohlo být realizováno pouze několik vstupů a výstupů a nebo i žádný.

Topologie modulu rádia z pohledu této specifikace je nejjednodušší možná. Je zde pouze jeden výstupní terminál přenosu zvuku přes USB, který má jako vstup nastaven vstupní terminál přijímač rádia.

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x04	Typ deskriptoru.
bInterfaceNumber	1B	0	Pořadové číslo rozhraní.
bAlternateSetting	1B	0	Identifikátor alternativní nastavení.
bNumEndpoints	1B	0	Počet koncových bodů v tomto rozhraní.
bInterfaceClass	1B	1	Třída rozhraní. (Audio)
bInterfaceSubClass	1B	1	Podtřída rozhraní. (Control device)
bInterfaceProtocol	1B	0	Vždy 0.
iInterface	1B	0	Index na textový řetězec.

Tabulka 4: Deskriptor řídicího rozhraní zvuku.

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x24	Typ deskriptoru.
bDescriptorSubType	1B	0x01	Pod typ deskriptoru. (Hlavička)
bcdADC	2B	0x0100	Verze USB Audio specifikace.
wTotalLength	2B	30	Celková délka deskriptorů tohoto rozhraní.
bInCollection	1B	1	Počet rozhraní pro odesílání zvuku.
baInterfaceNr(1)	1B	1	Index rozhraní pro odesílání zvuku.

Tabulka 5: Deskriptor řídicího rozhraní zvuku - hlavička.

Název pole	Délka	Hodnota	
bLength	1B	12	Délka deskriptoru.
bDescriptorType	1B	0x24	Typ deskriptoru.
bDescriptorSubType	1B	0x02	Pod typ deskriptoru. (Vstupní terminál)
bterminalID	1B	1	Identifikátor terminálu.
wTerminalType	2B	0x0710	Typ terminálu. (Přijímač rádia)
bAssocTerminal	1B	0	Přidružený terminál. (Nejedná se o zvukové propojení)
bNrChannels	1B	2	Počet zvukových kanálů.
wChannelConfig	2B	0x0003	Bitová mapa konfigurace kanálů.
iChannelNames	1B	0	Index na textový řetězec s názvem kanálů.
iTerminal	1B	0	Index na textový řetězec s popisem terminálu.

Tabulka 6: Deskriptor řídicího rozhraní zvuku - vstupní terminál.

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x24	Typ deskriptoru.
bDescriptorSubType	1B	0x03	Pod typ deskriptoru. (Výstupní terminál)
bterminalID	1B	2	Identifikátor terminálu.
wTerminalType	2B	0x0101	Typ terminálu. (Odesílání přes USB)
bAssocTerminal	1B	0	Přidružený terminál. (Nejedná se o zvukové propojení)
bSourceID	1B	1	Identifikátor připojeného vstupního terminálu.
iTerminal	1B	0	Index na textový řetězec s popisem terminálu.

Tabulka 7: Deskriptor řídicího rozhraní zvuku - výstupní terminál.

Řízení probíhá přes řídicí rozhraní zvuku (Audio control interface), které je vždy napojeno na koncový bod 0. V popisu tohoto rozhraní se skrývá i topologie. Skládá se z deskriptorů v tabulkách 4, 5, 6 a 7.

První deskriptor pouze určuje, že bude následovat popis rozhraní řízení zvuku třídy audio.

Následující deskriptor (tabulka 5) tvoří hlavičku třídně specifického popisu rozhraní. Specifikuje jednak použitou verzi USB Audio specifikace a také seznam rozhraní pro odesílání zvuku přes USB. Počet položek v tomto seznamu určuje hodnota pole bInCollection. Poté se na konec deskriptoru pro každé rozhraní přidá jedna položka. V tomto případě je použito pouze jediné rozhraní.

Zbývá dvojice deskriptorů (tabulka 6 a 7) popisuje vlastní topologii. Výstupní terminály mají vždy pole bSourceID, které se nastaví na hodnotu bTerminalType vstupního terminálu se kterým je propojen.

Dále následuje sada deskriptorů popisující rozhraní pro odesílání zvuku (audio streaming interface). Stejně jako řídicí rozhraní zde se začíná standardním deskriptorem roz-

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x04	Typ deskriptoru.
bInterfaceNumber	1B	1	Pořadové číslo rozhraní.
bAlternateSetting	1B	0	Identifikátor alternativní nastavení.
bNumEndpoints	1B	0	Počet koncových bodů v tomto rozhraní.
bInterfaceClass	1B	1	Třída rozhraní. (Audio)
bInterfaceSubClass	1B	2	Podtřída rozhraní. (Streaming)
bInterfaceProtocol	1B	0	Vždy 0.
iInterface	1B	0	Index na textový řetězec.

Tabulka 8: Deskriptor rozhraní odesílání zvuku.

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x04	Typ deskriptoru.
bInterfaceNumber	1B	1	Pořadové číslo rozhraní.
bAlternateSetting	1B	1	Identifikátor alternativní nastavení.
bNumEndpoints	1B	1	Počet koncových bodů v tomto rozhraní.
bInterfaceClass	1B	1	Třída rozhraní. (Audio)
bInterfaceSubClass	1B	2	Podtřída rozhraní. (Streaming)
bInterfaceProtocol	1B	0	Vždy 0.
iInterface	1B	0	Index na textový řetězec.

Tabulka 9: Deskriptor rozhraní odesílání zvuku.

hraní, který nese základní informace o rozhraní.

V tomto případě jsou použity hned dva deskriptory (tabulka 8 a 9). Liší se pouze ve dvou polích a to bAlternateSetting a bNumEndpoints. Druhé z těchto polí určuje počet koncových bodů použitých v rozhraní. Díky tomu, že je v prvním deskriptoru tento počet nulový, má host možnost zvolit toto alternativní nastavení vždy když nemá zájem o zvuková data. K identifikaci tohoto nastavení slouží hodnota v prvním ze zmíněných polí bAlternateSettings.



Název pole	Délka	Hodnota	
bLength	1B	7	Délka deskriptoru.
bDescriptorType	1B	0x24	Typ deskriptoru.
bDescriptorSubType	1B	1	Podtyp deskriptoru.
bTerminalLink	1B	2	Index výstupního terminálu.
bDelay	1B	1	Zpoždění v paketech.
wFormatTag	2B	1	Formát přenášených dat. (PCM)

Tabulka 10: Deskriptor formátu zvuku - hlavička.

Název pole	Délka	Hodnota	
bLength	1B	11	Délka deskriptoru.
bDescriptorType	1B	0x24	Typ deskriptoru.
bDescriptorSubType	1B	2	Podtyp deskriptoru.
bFormatType	1B	1	Typ formátu.
bNrChannels	1B	2	Počet kanálů.
bSubFrameSize	1B	2	Velikost jednoho pod-rámce v bytech.
bSubFrameResolution	1B	16	Počet platných bitů jednoho pod-rámce.
bSamFreqType	1B	1	Způsob určení vzorkovací frekvence. (Diskrétní)
tSamFreq(1)	3B	48000	Vzorkovací frekvence v Hz.

Tabulka 11: Deskriptor formátu zvuku.

V tabulce 10 a 11 jsou deskriptory popisující formátování a způsob přenosu zvukových dat. Formátování dat je specifikováno v [4]. Podle tohoto dokumentu se proud zvukových dat dělí do paketů, které se odesílají každou 1 ms. Paket je složen z rámců (frames), které reprezentují úroveň všech kanálů zachycených v jeden okamžik. Samotné úrovně jednotlivých kanálů se nazývají pod-rámce (subframes).

Deskriptor třídně specifické hlavičky v tabulce 10 nese kromě poli pro identifikaci sebe sama pouze tři zajímavá pole.

První je bTerminalLink. Identifikuje ke kterému výstupnímu terminálu se popis formátu formát vztahuje.

Pole bDelay je hodnota prodlevy mezi zachycením vzorku a jeho odesláním. Veličinou je počet paketů. Hodnota tohoto pole má význam pouze v případě simultánního záznamu zvuku z více zvukových karet a podobně. V případě přijímače rádia na ní nezáleží.

Posledním polem je wFormatTag, které určuje, že hodnoty jednotlivých vzorků budou ve formátu PCM jako čísla se znaménkem. I zde nabízí USB Audio specifikace značnou volnost a kromě formátu PCM je možné přenášet zvuk komprimovaný různými kodeky. Detailnější specifikace je v [4].

Následující deskriptor, vyobrazený v tabulce 11, blíže specifikuje parametry PCM formátu. Toto specifikuje hodnota pole `bFormatType`.

Pole určuje, jak název napovídá, `bNrChannels` určuje počet kanálů. Pole `bSubframeSize` určuje velikost jednoho pod-rámce (vzorku) v Bytech. Zároveň pole `bSubFrameResolution` určuje počet platných bitů v pod-rámci.

Konec deskriptoru je věnován specifikaci vzorkovacích frekvencí. Pomocí hodnoty v poli `bSamFreqType` se určí způsob specifikace frekvencí. V případě modulu je použita pouze jediná diskrétní frekvence, která je specifikována v poli `tSamFreq(1)`. Je možné zde specifikovat více diskrétních frekvencí, ze kterých si poté hostitel vybírá a nebo specifikovat interval.

Název pole	Délka	Hodnota	
<code>bLength</code>	1B	7	Délka deskriptoru.
<code>bDescriptorType</code>	1B	0x25	Typ deskriptoru.
<code>bEndpointAddress</code>	1B	0x81	Číslo koncového bodu.
<code>bmAttributes</code>	1B	0b00000001	Bitová mapa s atributy.
<code>wMaxPacketSize</code>	2B	384	Maximální velikost paketu v bytech.
<code>bInterval</code>	1B	1	Interval Odesílání.
<code>bRefresh</code>	1B	0	Frekvence synchronizace. (nepoužito)
<code>bSynchAddress</code>	1B	0	Adresa synchronizačního koncového bodu. (Nepoužito)

Tabulka 12: Deskriptor koncového bodu odesílání zvuku.

V tabulce 12 je deskriptor koncového bodu popisující koncový bod odesílání zvuku (audio streaming endpoint). Prvním zajímavým polem je adresa koncového bodu v poli `bEndpointAddress`. Jako adresa je použito číslo koncového bodu s tím, že v případě vstupního koncového bodu (směr do hostitele) se sedmý (nejvyšší) bit nastaví na 1. U opačného směru tento bit zůstává nulový.

Pole `bmAttributes` obsahuje bitovou mapu atributů koncového bodu. Význam bitů 0 a 1 je následující:

- 00 - Řídící koncový bod
- 01 - Izochronní koncový bod
- 10 - Hromadný koncový bod
- 11 - Koncový bod přerušení

Zbylé bity popisují způsob synchronizace izochronních koncových bodů, která ale není použita. Jejich popis je v kapitole 5.12.4 USB specifikace [2].

Pole `wMaxPacket size` určuje největší možnou velikost paketu. V tomto případě se spočítá z deskriptoru v tabulce 11 jako  $bNrChannels \cdot bSubFrameSize \cdot \frac{tSamFreq(1)}{1000}$ . Vypočtenou hodnotu je nutné vždy zaokrouhlit nahoru. Například v případě vzorkovací frekvence 44,1 kHz bude ve většině paketů 44 rámců, ale každý desátý jich bude mít 45.

Pole `bInterval` určuje periodu čtení dat z koncového bodu. Hodnota 1 znamená, že se bude hostitel pokoušet číst data z koncového bodu každou 1 ms. Zbývající dvě pole se vztahují k synchronizaci, která nebyla použita a tudíž jsou nastavena na hodnotu 0.

Název pole	Délka	Hodnota	
<code>bLength</code>	1B	9	Délka deskriptoru.
<code>bDescriptorType</code>	1B	0x05	Typ deskriptoru.
<code>bDescriptorSubType</code>	1B	1	Podtyp deskriptoru.
<code>bmAttributes</code>	1B	0b00000000	Bitová mapa s atributy.
<code>bLockDelayUnits</code>	1B	0	Jednotky prodlevy stabilizace synchronizace. (Nepoužito).
<code>wLockDelay</code>	2B	0	Doba prodlevy stabilizace synchronizace. (Nepoužito)

Tabulka 13: Deskriptor koncového bodu odesílání zvuku specifický pro danou třídu.

Posledním deskriptorem popisujícím přenos zvuku je třídě specifický deskriptor koncového bodu v tabulce 13. Za běžnými uvozujícími poli se nachází bitová mapa s atributy `bmAttributes`. Význam bitů je následující:

- Bit 0 - Koncový bod podporuje žádost o počáteční nastavení vzorkovací frekvence.
- Bit 1 - Koncový bod podporuje dynamickou změnu vzorkovací frekvence za běhu.
- Bit 2-6 - Vyhrazeno - musí být nulové.
- Bit 7 - Pokud je nastaveno musí se přenášená data vždy doplnit nulami, tak aby měl každý paket velikost `wMaxPacketSize`.

Dvě pole na konci deskriptoru umožňují specifikovat dobu potřebnou pro stabilizaci synchronizace. Synchronizace koncového bodu není použita takže obě hodnoty jsou nulové.

### 3.1.9 Toto asi vyhodit...

Deskriptory jsou binární struktury takže je nutné vzít v potaz endianitu. U USB je to vždy little-endian, to znamená, že se číslo odesílá od nejméně významného bytu po nejvíce významný. Například 0x1234 se přenese nejdříve 0x34 a poté 0x12. Mikrokontrolér PIC32 s dodávaným kompilátor XC32 používá taktéž little-endian, je z hlediska případné přenositelnosti výhodné implementovat jako pole typu **char** viz ukázka 3.1.9.

---

```
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.

## 3.2 Microchip Harmony framework

TODO:

- Odkaz a bug se špatnou inicializací a následným dělení nulou.
- Popsat chybu kdy nedojde k povolení AS EP na žádost set alt setting 1
- Popsat žádost a potom implementaci přes vendor EP
- Zmínit že jsem došel k té samé chybě když jsem s senaži povolovat EP sám.
- zmínit že s trvale povoleným EP to funguje ale přenese se pouze každý druhý paket. a doplnit screenshotem z analyzáru

## 3.3 Vlastní implementace

TODO:

- Popsat žádosti nutné pro konfiguraci a běh
- zmínit ping-pong buffery a rozhraní MCU pro USB
- Popsat rozhraní vzniklého USB stacku ??

### 3.4 USB I<sup>2</sup>C tunel

TODO:

- základní filozofie - vendor interface, 2x bulk EP
- popsat formáty zpráv pro oba směry (zmínit endiianitu)
- chyby v křemíku - problém možného dvojí zápisu po návratu z přerušení a nefunkčnost dvou pinů při zapnutí I<sup>2</sup>C modulu.

### 3.5 Omezení

TODO

- Nemožnost dostat se v suspendu do 0,5 mA
- Test mód neimplementován. Nutný pouze pro získání USB loga.

## **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](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](http://www.usb.org/developers/docs/devclass_docs/audio10.pdf)
- [4] Gal Ashour, Billy Brackenridge, Oren Tirosh, Altec Lansing, Craig Todd, Remy Zimmermann, Geert Knapen. *Universal Serial Bus Device Class Definition for Audio Data Formats* [online] 1998-03-18 [2016-02-20] [http://www.usb.org/developers/docs/devclass\\_docs/frmts10.pdf](http://www.usb.org/developers/docs/devclass_docs/frmts10.pdf)
- [5] USB Implementers' Forum. *Universal Serial Bus Language Identifiers (LANGIDs): Version 1.0* [online] 2000-03-26 [2016-02-09] [http://www.usb.org/developers/docs/USB\\_LANGIDs.pdf](http://www.usb.org/developers/docs/USB_LANGIDs.pdf)
- [6] 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>
- [7] 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>
- [8] 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>
- [9] 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>
- [10] 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>
- [11] 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>

- 
- [12] 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>
  - [13] 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>
  - [14] 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>
  - [15] *TAS1020B USB Streaming Controller* [online] 2011-05 [2015-12-29] <http://www.ti.com/lit/ds/symlink/tas1020b.pdf>
  - [16] 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>
  - [17] USB.org *Getting a Vendor ID* [online] [2016-01-16] <http://www.usb.org/developers/vendor/>

## **A Schéma zapojení modulu**

Na štorc