

# **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 cenné rady a 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>Tuner</b>	<b>10</b>
3.1	Zvukové rozhraní I <sup>2</sup> S . . . . .	10
3.2	Ovládací rozhraní I <sup>2</sup> C . . . . .	10
<b>4</b>	<b>USB</b>	<b>12</b>
4.1	Stručný úvod do full-speed USB 2.0 . . . . .	12
4.2	Microchip Harmony framework . . . . .	25
4.3	Vlastní implementace USB . . . . .	25
4.4	USB I <sup>2</sup> C tunel . . . . .	30
4.5	Omezení . . . . .	30
<b>5</b>	<b>Knihovna</b>	<b>31</b>
5.1	Nízko úrovněvé funkce . . . . .	31
5.2	Středně úrovněvé funkce . . . . .	31
5.3	Vysoko úrovněvé funkce . . . . .	31
5.4	Překlad . . . . .	31
<b>6</b>	<b>Závěr</b>	<b>32</b>
<b>7</b>	<b>Reference</b>	<b>33</b>
	<b>Přílohy</b>	<b>34</b>

## Seznam tabulek

1	Popis I <sup>2</sup> S signálů. . . . .	10
2	I <sup>2</sup> C adresy. . . . .	11
3	Druhy USB přenosů. . . . .	12
4	Deskriptor zařízení. . . . .	15
5	Deskriptor konfigurace. . . . .	17
6	Deskriptor řídicího rozhraní zvuku. . . . .	18
7	Deskriptor řídicího rozhraní zvuku - hlavička. . . . .	18
8	Deskriptor řídicího rozhraní zvuku - vstupní terminál. . . . .	19
9	Deskriptor řídicího rozhraní zvuku - výstupní terminál. . . . .	19
10	Deskriptor rozhraní odesílání zvuku. . . . .	20
11	Deskriptor rozhraní odesílání zvuku. . . . .	20
12	Deskriptor formátu zvuku - hlavička. . . . .	21
13	Deskriptor formátu zvuku. . . . .	21
14	Deskriptor koncového bodu odesílání zvuku. . . . .	22
15	Deskriptor koncového bodu odesílání zvuku specifický pro danou třídu. . . . .	23
16	Deskriptor rozhraní USB - I <sup>2</sup> C tunelu. . . . .	24
17	Deskriptor koncových bodů USB - I <sup>2</sup> C tunelu. . . . .	24
18	Hlavička USB požadavků. . . . .	28



## Seznam obrázků

1	Blokové schéma TAS1020b. (Převzato z [15]) . . . . .	8
2	Blokové schéma zapojení. . . . .	9
3	Časový diagram I <sup>2</sup> S přenosu. (Převzato z [18]) . . . . .	11
4	USB endpointy a funkce. . . . .	13
5	Formát deskriptoru USB bufferu předaného do USB modulu. (Převzato z [12].) . . . . .	26
6	Formát deskriptoru USB bufferu předaného do aplikace. (Převzato z [12].) . . . . .	26
7	Zjednodušený diagram stavů USB zařízení. . . . .	29

## **Seznam výpisů zdrojového kódu**

## 1 Úvod

## 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 diskretních 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.

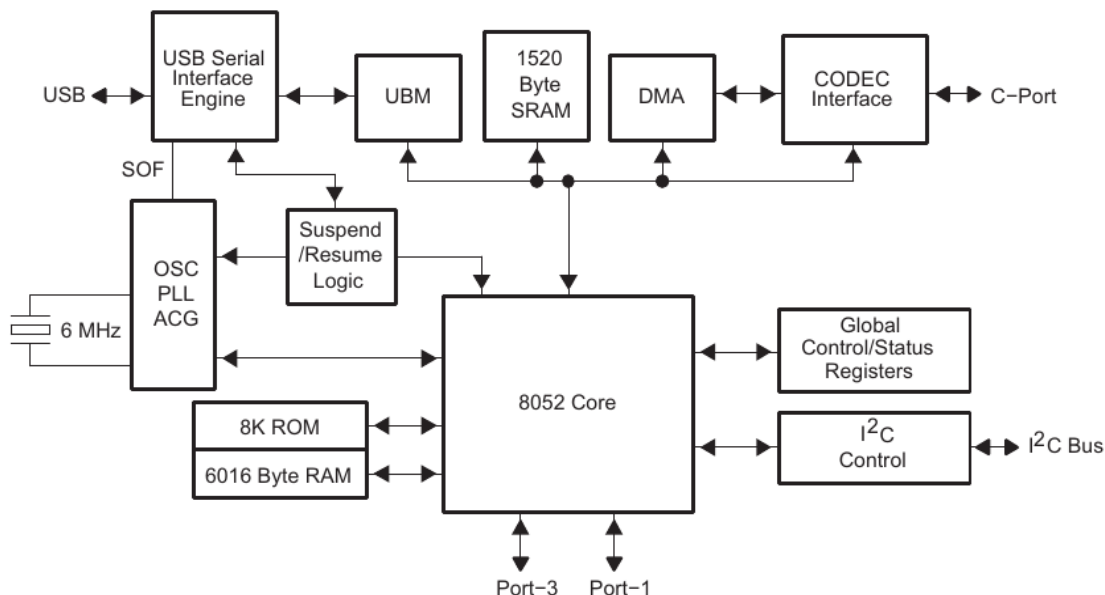
### 2.3.1 TAS1020b

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.

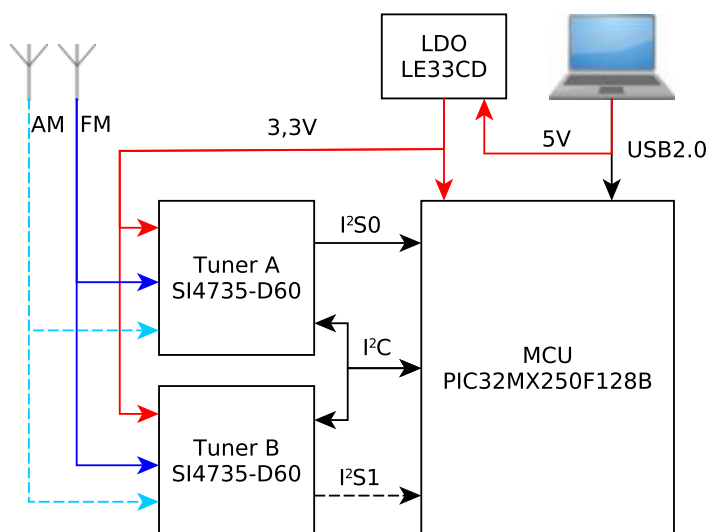
Výrobce poskytuje k tomuto MCU knihovnu Microchip Library for applications, která mimo jiné obsahuje implementaci USB audio 1.0. Navíc jedním ze vzorových projektů u této knihovny je i USB mikrofón, který řeší přenos zvuku do počítače.

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 Tuner

Jak jsem zmínil v kapitole 2.4

- nepoužití přerušení a důsledky - nutnost polingu
- nějak vkusně s ohledem na předchozí kapitolu vyzdvihnout použití I<sup>2</sup>C - jendou větou na začátek

#### 3.1 Zvukové rozhraní I<sup>2</sup>S

MCU	Směr	Tuner	Význam
SDI	⇒	DOUT	Datový signál
SCK	⇒	DCLK	Hodiny
SS	⇐	DFS	Signál určení kanálu

Tabulka 1: Popis I<sup>2</sup>S signálů.

Rozhraní I<sup>2</sup>S je v případě jednosměrného přenosu v podstatě rozhraní SPI doplněné o další datový signál. Význam signálů včetně odpovídajících názvů pinů na MCU a tuneru shrnuje tabulka 1. V případě I<sup>2</sup>S se nejedná o sběrnici. Vždy komunikují právě dvě zařízení kde jedno je master a druhé slave. Master vždy vysílá datový signál a signál přepínání kanálů. Datový signál vysílá samozřejmě zřízení, které je zdrojem zvuku. Je možný i obousměrný přenos, potom je nutné použití dvou datových linek.

Tuner umí pracovat pouze v režimu slave, takže hodinový signál a signál přepínání kanálů je generován z MCU. Vzorkování zvuku v tuneru je řízeno jeho vlastním interním oscilátorem. Díky tomuto pravděpodobně občas dojde k zahození některého vzorku pokud MCU čte pomaleji než tuner provádí vzorkování, a nebo dojde k přečtení náhodných dat v opačném případě. Nicméně v praxi jsem s tímto nezaznamenal žádný problém.

Vyjma standardního formátu dat popsaného v I<sup>2</sup>S specifikace [18] vzniklo ještě několik dalších formátů. Použil jsem standardní formát, tak jak je zobrazen na obrázku 3.

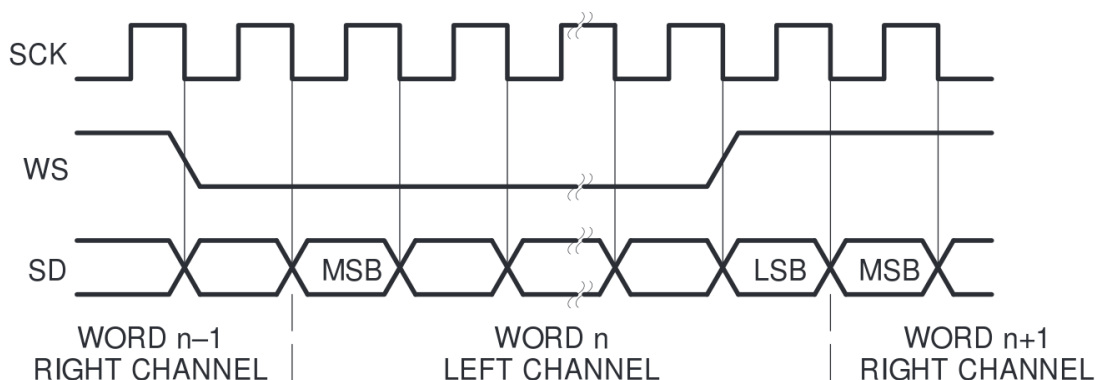
TODO:

- Umravnit pozicování obrázku
- Problém synchronizace hodin - presunout do USB jako samostatnou podkapitolu a nebo spáchat samostatnou kapitolu přeposílání zvuku

#### 3.2 Ovládací rozhraní I<sup>2</sup>C

Ovláddání tuneru se provádí přes rozhraní I<sup>2</sup>C . Toto rozhraní je součástí drtivé většiny mikrokontrolérů, není ho třeba blíže popisovat. Tuner se do režimu I<sup>2</sup>C přepne připojením pinů GPO1 na 1 GPO2 na 0.





Obrázek 3: Časový diagram I²S přenosu. (Převzato z [18])

SEN	Adresa čtení	Adresa zápisu
0	0x23	0x22
1	0xC7	0xC6

Tabulka 2: I²C adresy.

Základní informací nezbytnou pro komunikace s tunerem je I²C adresa. Tuner umožňuje změnu adresy pomocí změny úrovně na pinu SEN. Adresy jsou v tabulce 2.

Tuner se ovládá zapisováním příkazů a případným čtením odpovědí. Dále je zde dvojice příkazů pro čtení a zápis proměnných. (properties) tuneru. Popis všech příkazů a proměnných je v programovací příručce k tuneru [7].

TODO:

- minimálně obecný formát příkazů
- čtení a zápis se nikdy neprolíná
- popsat že se ovládá pomocí příkazů a properties
- a přepsat sem použité příkazy ?
- možná přidat kapitolku se spuštěním tuneru - možná jenom flowchart a možná ho hodit do přílohy - nechme to na konec kdyby nevycházel rozsah tak se popíše textově

### 3.2.1 Čtení RDS z tuneru

TODO:

- vyčtení z tuneru
- dekodování základních informací- třeba radio text

## 4 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.

### 4.1 Stručný úvod do full-speed USB 2.0

#### 4.1.1 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.

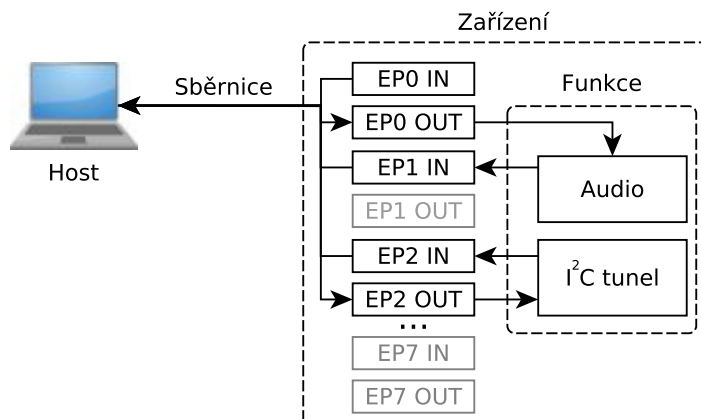
Ve specifikaci je také zohledněn fakt, že zařízení zpravidla nezastává pouze jedinou funkci. To je konec konců případ i tohoto modulu. Obsahuje dvě funkce - zvukovou kartu a I<sup>2</sup>C tunel pro komunikaci s tunery.

#### 4.1.2 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 3: 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 3. Výjimkou je vstupní a výstupní koncový bod nula. Tyto vždy slouží pro řídicí



Obrázek 4: 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 3.

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 4.

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 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) 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].

#### 4.1.3 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 4: Deskriptor zařízení.

#### 4.1.4 Deskriptor zařízení

V tabulce 4 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.

### 4.1.5 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 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).

### 4.1.6 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 5: Deskriptor konfigurace.

#### 4.1.7 Deskriptor konfigurace

Jak je vidět z tabulky 5, tento deskriptor obsahuje informace pro identifikaci konfigurace. A to zejména index *iConfiguration*. To je hodnota, kterou poté pošle host do zařízení v požadavku o přidělení konfigurace. Dále následuje pole *wTotalLength* s celkovou délkou konfigurace. 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 7 signalizuje, že zařízení chce využívat mechanismus vlastního probuzení a informování hostitele o události. Modul má oba atributy nastaveny na hodnotu 0.

#### 4.1.8 Deskriptory konfigurace vztahované k 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 6: Deskriptor řídicího rozhraní zvuku.

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x24	Typ deskriptoru.
bDescriptorSubType	1B	0x01	Podtyp 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 7: Deskriptor řídicího rozhraní zvuku - hlavička.



Název pole	Délka	Hodnota	
bLength	1B	12	Délka deskriptoru.
bDescriptorType	1B	0x24	Typ deskriptoru.
bDescriptorSubType	1B	0x02	Podtyp 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 8: Deskriptor řídicího rozhraní zvuku - vstupní terminál.

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x24	Typ deskriptoru.
bDescriptorSubType	1B	0x03	Podtyp 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 9: 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 6, 7, 8 a 9.

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

Následující deskriptor (tabulka 7) 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 8 a 9) 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 10: 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 11: Deskriptor rozhraní odesílání zvuku.

hraní, který nese základní informace o rozhraní.

V tomto případě jsou použity hned dva deskriptory (tabulka 10 a 11). 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 12: 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 podrámce v bytech.
bSubFrameResolution	1B	16	Počet platných bitů jednoho podrámce.
bSamFreqType	1B	1	Způsob určení vzorkovací frekvence. (Diskrétní)
tSamFreq(1)	3B	48000	Vzorkovací frekvence v Hz.

Tabulka 13: Deskriptor formátu zvuku.

V tabulce 12 a 13 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 podrámce (subframes).

Deskriptor třídně specifické hlavičky v tabulce 12 nese kromě polí pro identifikaci sebe sama pouze tři zajímavá pole.

První je bTerminalLink. Identifikuje, ke kterému výstupnímu terminálu se popis formátu 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 13, blíže specifikuje parametry PCM formátu. Toto specifikuje hodnota pole `bFormatType`.

Jak název napovídá, pole `bNrChannels` určuje počet kanálů. Pole `bSubframeSize` obsahuje velikost jednoho podrámcu (vzorku) v Bytech. Zároveň pole `bSubFrameResolution` nese počet platných bitů v podrámcu.

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 14: Deskriptor koncového bodu odesílání zvuku.

V tabulce 14 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 `wMaxPacketSize` určuje největší možnou velikost paketu. V tomto případě se spočítá z deskriptoru v tabulce 13 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 15: Deskriptor koncového bodu odesílání zvuku specifický pro danou třídu.

Posledním deskriptorem popisujícím přenos zvuku je třídně specifický deskriptor koncového bodu v tabulce 15. 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.

#### 4.1.9 Deskriptory konfigurace vztažené k USB I<sup>2</sup>C tunelu

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x04	Typ deskriptoru.
bInterfaceNumber	1B	2	Číslo rozhraní.
bAlternateSettings	1B	0	Identifikátor alternativního nastavení.
bNumEndpoints	1B	2	Počet koncových bodů v tomto rozhraní.
bInterfaceClass	1B	255	Třída rozhraní (Vendor specific).
bInterfaceSubClass	1B	0	Podtřída rozhraní.
bInterfaceProtocol	1B	255	Protokol rozhraní (Vendor specific).
iInterface	1B	0	Index řetězce s popisem rozhraní (Nepoužit).

Tabulka 16: Deskriptor rozhraní USB - I<sup>2</sup>C tunelu.

Deskriptor rozhraní USB - I<sup>2</sup>C tunelu v tabulce 16 je podobný jako zvuková rozhraní. Liší se samozřejmě třídou zařízení, kde třída 255 určuje, že se nejedná o žádné ze standardizovaných rozhraní, ale je definováno výrobcem. Stejný význam má protokol číslo 255.

Název pole	Délka	Hodnota	
bLength	1B	9	Délka deskriptoru.
bDescriptorType	1B	0x05	Typ deskriptoru.
bEndpointAddress	1B	0x82 / 0x02	Číslo koncového bodu.
bmAttributes	1B	0b00000010	Bitová mapa s atributy.
wMaxPacketSize	2B	32	Maximální velikost paketu v bytech.
bInterval	1B	0	Interval Odesílání. (Nemá význam)
bRefresh	1B	0	Frekvence synchronizace. (Nemá význam)
bSynchAddress	1B	0	Adresa synchronizačního koncového bodu. (Nemá význam)

Tabulka 17: Deskriptor koncových bodů USB - I<sup>2</sup>C tunelu.

V tabulce 17 jsou oba deskriptory koncových bodů pro USB I<sup>2</sup>C tunel. Liší se pouze v adrese. Pole bmAttributes obsahuje pouze informaci o tom že se jedná o koncový bod hromadného přenosu dat (bulk endpoint). Poslední tři pole mají v případě tohoto druhu koncového bodu vždy hodnotu 0.

Maximální velikost paketu wMaxPacketSize může být 8, 16, 32 nebo 64 B (viz. [2] kap. 5.8.3). Vzhledem k velikosti nejdelšího možného příkazu a odpovědi odesílaného přes I<sup>2</sup>C . Jsem zvolil velikost 32 B.

## 4.2 Microchip Harmony framework

Harmony framework obsahuje mimo jiné také ovladače pro USB včetně USB Audio 1.0. Součástí je i vzorový projekt pro USB reproduktor, ze kterého jsem původně vycházel.

Po té co jsem projekt upravil tak aby zvuková data nepřijímal ale odesílal se začal MCU restartovat. Z fóra firmy Microchip (<http://www.microchip.com/forums/FindPost/827487>) jsem zjistil, že jde o již nahlášený problém chybné inicializace v důsledku které dochází k dělení nulou a k resetu MCU. Na fóru je také uveřejněno jak tuto chybu opravit.

Opravením této chyby jsem se dostal k odesílání zvuku. I když modul správně prošel enumerací a konfigurací do hostitele se přenášelo ticho. Pomocí USB analyzátoru jsem zjistil, že modul od hostitele dostává IN tokeny, ale posílá pouze pakety nulové délky jako by neměl žádná data k odeslání.

Rozhodl jsem se implementovat ovladač USB Audia jako vendor endpoint (koncový bod specifikovaný výrobcem). Pro potřeby modulu stačí pouze zajistit odesílání zvuku přes izochronní koncový bod a reagovat na požadavek změny alternativního nastavení funkce povolením nebo zakázáním Izochronního koncového bodu. Došel jsem ke zcela totožnému chování modulu jako s původním ovladačem.

Upravil jsem vlastní ovladač tak, aby izochronní koncový bod byl povolen neustále. Po této upravě se začal přenášet zvuk, ale implementace vendor endpointu je pravděpodobně příliš pomalá takže přibližně každý druhý paket byl následován paketem nulové délky.

TODO: Možná doplnit screenshotem

## 4.3 Vlastní implementace USB

Kvůli zmiňovaným problémům s Harmony frameworkem, které se mi nepodařilo vyřešit, jsem se rozhodl napsat ovladač v MCU sám. Ke zprovoznění modulu je potřeba, aby MCU podporoval pouze několik požadavků z hosta a implementovat jednoduchý stavový stroj.

### 4.3.1 USB modul v MCU

Dokumentace USB v MCU se nachází v obecném manuálu [12] pro celou řadu těchto 32 bitových mcu. Detaily pro konkrétní typ jsou v manuálu [8]. Pravděpodobně díky existenci Harmony frameworku je dokumentace relativně stručná.

USB modul v MCU PIC32MX je velmi podobný USB modulu v 8 bitovém MCU PIC16F1454 na kterém jsem se snažil původně modul tuneru postavit. Podle USB specifikace nesmí být na sběrnici přivedeno žádné napětí pokud není sběrnice napájena z hostitele. U zařízení napájených z USB sběrnice toto není třeba řešit. MCU PIC32 má vyhrazený vstup VBUS právě pro snímání napětí na USB sběrnici. I když je ho možné pomocí konfiguračních slov deaktivovat je nutné jej použít. Pokud na tento vstup není přivedeno napětí, USB modul v MCU nepracuje.

Address Offset +0

31					26	25									16	15											8	7	6	5	4	3	2	1	0
—					BYTE_COUNT<9:0>											—					UOWN	DATA0/1	KEEP	NINC	DTS	BSTALL	—								

Address Offset +4

31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					</
----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

Obrázek 5: Formát deskriptoru USB bufferu předaného do USB modulu. (Převzato z [12].)

Address Offset +0

31					26	25									16	15											8	7	6	5	4	3	2	1	0
—					BYTE_COUNT<9:0>											—					UOWN DATA0/1	PID<3:0>					—								

Address Offset +4

31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Obrázek 6: Formát deskriptoru USB bufferu předaného do aplikace. (Převzato z [12].)

Vlastní konfigurace USB modulu je v případě USB zařízení v celku jednoduchá. Bitem USBPWR v registru U1PWRC se provede spuštění USB modulu.

Dále je potřeba do registrů U1BDTP1-3 nastavit dolní tři byty ukazatele na tabulku deskriptorů bufferů. Každému koncovému bodu pro každý směr jsou přiřazeny dva záznamy v této tabulce, lichý a sudý. Tvoří takto dvouprvkovou FIFO frontu. Každý záznam v této tabulce nese ukazatel na buffer pro příjem nebo odesílání dat a další nezbytné informace. Tabulka může být umístěna kdekoli v zapisovatelné paměti MCU, ale její adresa musí být zarovnaná na 512 B. Jádro MCU používá rozdílné adresy pro přístup k datům z programu (virtuální adresy) a přístup k datům z hardwarových modulů (fyzické adresy). Adresu ukládanou do registrů U1BDTP je tudíž potřeba přeložit. To se provede logickým součinem s hodnotou 0x1FFFFFFF.

Dále je potřeba inicializovat samotnou tabulku deskriptorů bufferů. Jak je patrné z obrázků 5 a 6 význam dat v jednotlivých záznamech tabulky je mírně odlišný v případě že tabulku předáváme do USB modulu a v případě že USB modul zpracoval token pro daný koncový bod a tabulku vrátil aplikaci. Význam jednotlivých polí je následující:

- **BYTE\_COUNT** - Počet bytů k přijetí nebo odeslání.
- **UOWN** - Pokud je nastaven na 1, buffer patří USB modulu a nelze jej měnit. Po zpracování paketu bude tento bit nastaven USB module zpět na 0.



- **DATA0/1** - Určuje jestli se paket odešle s PID DATA0 nebo DATA1. V případě příjmu určí jestli se očekává paket s PID DATA0 nebo DATA1.
- **KEEP** - Pokud je nastaven na 1, USB modul si tento buffer ponechá napořád.
- **NINC** - Pokud je nastaven na adresa DMA nebude inkrementována.
- **DTS** - Pokud je nastaven na 1, USB modul bude při příjmu ignorovat pakety kde PID (DATA0/1) neodpovídá hodnotě pole DATA0/1. Pokud je nastaven na 0, přijme všechny pakety.
- **BSTALL** - Pokud je nastaven na 1, byl přijat nebo bude vyslán STALL.
- **PID** - Druh přijatého tokenu.
- **BUFFER\_ADDRESS** - Fyzická adresa na buffer s data k odeslání nebo příjmu.

Jak je patrné z předchozího výčtu je nezbytné inicializovat přinejmenším bit UOWN na hodnotu 0.

Po tomto kroku je možné spustit USB modul nastavením bitu USBEN v registru U1CON na 1.

Aplikace si musí udržovat informace o tom jestli má používat lichý nebo sudý buffer. USB modul umožňuje resetovat všechny koncové body v obou směrech tak, aby znova začaly sudým bufferem, nastavením bitu PBRST v registru U1CON na 1.

Ještě je nutné nakonfigurovat používané koncové body v registrech U1EP0 - U1EP15. Nastavením bitů EPTXEN a nebo EPRXEN odesílání a nebo příjem, pokud se nejedná o koncový bod pro řídicí přenosy je dobré zakázat příjem SETUP tokenů nastavením bitu EPCONDIS. A pokud se nejedná o koncový bod pro isochronní přenos dat povolit handshake nastavením bitu EPHSK.

Pro příjem a odesílání stačí nastavit příslušný záznam v tabulce deskriptorů bufferů s nastavením bitu UOWN jej předat do USB modulu. Po vyřízení příjmu nebo odesílání USB modul nastaví bit TRNIF v registru U1IR a současně nastaví v registru U1STAT tato pole:

- **ENDPT** - Číslo koncového bodu 0-15.
- **DIR** - Směr. Pokud má hodnotu 1 poslední transakce bylo odesílání (IN transakce), pokud 0 poslední transakce byla příjem (OUT transakce).
- **PPBI** - Pokud má hodnotu 1 byl vyřízen lichý buffer, pokud 0, byl vyřízen sudý buffer.
- **2 bity mezera.**

Pakliže je hodnota tohoto registru posunuta od dva bity doprava, je ji možné přímo použít jako index v tabulce deskriptorů bufferů.

Při příjmu setup tokenu (zahájení řídicí transakce) dojde k nastavení bitu PKTDIS registru U1CON. Dokud je tento bit nastaven je zastaveno zpracování všech tokenů. Po připravení odpovědi je nutné tento bit nastavit na hodnotu 0.

Aplikace by dále měla monitorovat v registru U1CON tyto bity: (Více v kapitole 4.3.3.)

- IDLEIF - Zařízení bylo suspendováno.
- URSTIF - Zařízení bylo resetováno.

#### 4.3.2 Standardní USB požadavky

Název pole	Délka	
bmRequestType	1B	Bitová mapa s obecným popisem požadavku.
bRequest	1B	Specifikace požadavku.
wValue	1B	Obecná hodnota.
wIndex	1B	Obecný index.
wLength	1B	Velikost dat, které je nutné dále přijmou nebo odeslat.

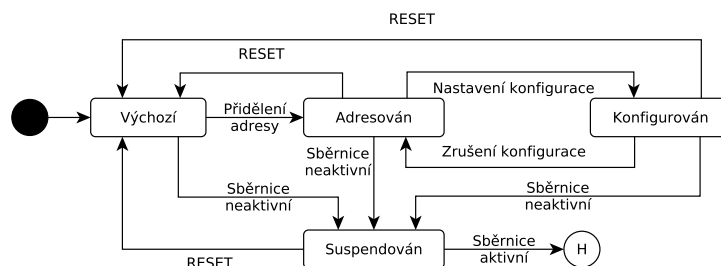
Tabulka 18: Hlavička USB požadavků.

Standardní požadavky na zařízení jsou detailně popsány v [2] v kapitole 9.3. Implementoval jsem pouze požadavky nutné k funkci modulu rádia.

Standardní požadavky odesílané z hostitele mají vždy stejnou hlavičku popsanou v tabulce 18. Význam polí wValue a wIndex je vždy specifický danému požadavek. Význam bitů pole bmRequestType je následující:

- Bit 7: Směr případné datové části.
  - 1 - Z hostitele do zařízení,
  - 0 - Ze zařízení do hostitele.
- Bity 6-5: Typ požadavku.
  - 0 - Standardní požadavek.
  - 1 - Požadavek specifický pro danou třídu zařízení.
  - 2 - Výrobcem specifikovaný požadavek.
  - 3 - Vyhrazeno.
- Bity 4-0: Příjemce požadavku.
  - 0 - Zařízení.
  - 1 - Rozhraní.
  - 2 - Koncový bod.
  - 3 - Jiný.
  - 4-31 - Vyhrazeno.

Byly implementovány následující požadavky: (V závorkách jsou uvedeny hodnoty bRequest)



Obrázek 7: Zjednodušený diagram stavů USB zařízení.

- **GET\_STATUS (0):** Odpovědí na tento požadavek jsou dva byty. Předposlední bit určuje jestli se zařízení smí probudit když je suspendováno (remote wakeup). Toto není podporováno, bit je vždy nula. Poslední nejnižší bit určuje jestli mám zařízení vlastní napájení. Tento bit je taktéž vždy nula. Ostatní bity jsou rezervovány a jsou vždy nulové.
- **SET\_ADDRESS (5):** V poli wValue zařízení obdrží novou adresu.
- **GET\_DESCRIPTOR (7):** Hostitel žádá o deskriptor. V horním byte wValue je typ deskriptoru, v dolním byte index deskriptoru. wIndex je buď nula nebo specifikuje ID jazyka. wLength určuje maximální délku odpovědi. Hostitel může požadovat pouze začátek deskriptoru.
- **GET\_CONFIGURATION (8):** Hostitel žádá o index aktuálně používané konfigurace. Ta je odeslána jako jeden byte.
- **SET\_ADDRESS (5):** V poli wValue zařízení obdrží index nové konfigurace.
- **SET\_INTERFACE (11):** V poli wValue zařízení obdrží index nového alternativní nastavení pro rozhraní s indexem v poli wIndex. (Implementováno v ovladači USB audio.)

#### 4.3.3 Stavy USB zařízení

Na obrázku 7 je stavový diagram USB zařízení. Diagram je zjednodušen pouze na stavy důležité pro modul.

- **Výchozí:** Stav po zapnutí a inicializaci zařízení. Při vstupu do tohoto stavu se nastaví adresa zařízení na výchozí hodnotu 0.
- **Adresován:** Do tohoto stavu se přejde po zpracování požadavku nastavení adresy. Veškerá další komunikace bude probíhat na této adrese. Ovšem adresu do registru MCU U1ADDR je nutné nastavit až po dokončení transakce - odeslání paketu nulové délky.

- **Konfigurován:** Do tohoto stavu se přejde na základě požadavku nastavení konfigurace.
- **Suspendován:** K suspendování zařízení může dojít obecně kdykoliv. Do tohoto stavu se přejde pokud na sběrnici není zaznamenána aktivita po dobu 3 ms. Toto detekuje hardware MCU. V tomto stavu musí zařízení snížit svoji spotřebu na nejvýše 0,5 mA.

Dále je potřeba počítat s faktem, že zařízení může být kdykoliv resetováno a vrátí se tak do Výchozího stavu, není stanoveno pořadí v jakém budou žádosti přicházet a kterákoli transakce může být předčasně ukončena vysláním dalšího setup tokenu.

#### 4.3.4 Rozhraní implementace USB

Implementace se nachází v projektu firmware ve složce drv.

V souboru config.h se nastaví počet konfigurací, celkový počet deskriptorů řetězců, nejvyšší použitý koncový bod a specifikuje se jestli má zařízení vlastní napájení.

V souboru usb.h je vlastní rozhraní.

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

TODO:

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

#### 4.5 Omezení

TODO

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

## **5 Knihovna**

DFD diagram asi není rozumný, ale určitě sem minimálně nakreslit rozvrstvení včetně vrstev v tuneru.

Dotaz asi na vedoucího - Přepisovat dokumentaci do češtiny a nebo se na ni jenom odkázat a omezit se na souhrn funkcí s vhodnými příklady.

### **5.1 Nízko úrovněvé funkce**

### **5.2 Středně úrovněvé funkce**

### **5.3 Vysoko úrovněvé funkce**

#### **5.3.1 RDS dekodér**

### **5.4 Překlad**

## 6 Závěr

TODO:

- Zmínit že příjem na PCB je mnohem kvalitnější než v nepáživém poli a změřit to pomocí SNR

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/>
  - [18] Philips Semiconductors *I<sup>2</sup>S bus specification* [online] 05-06-1996 [2016-02-27] <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00000545.pdf>