VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Výukový modul s FM rádiem

Training Module with FM Radio

2014 Patrik Štihel

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Patrik Štihel**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Výukový modul s FM rádiem
Training Module with FM Radio

Zásady pro vypracování:

Navrhněte a realizujte výukový modul pro vývojový KIT využívaný v předmětu Architektury počítačů a paralelních systémů. Modul bude obsahovat digitálně řízený obvod FM rádia. Tento obvod bude připojen na řídící sběrnici a jeho výstup bude napojen na reproduktor, případně na sluchátka.

1. Stručně popište AVR KIT používaný při výuce předmětu Architektury počítačů a paralelních systémů.
2. Vyberte vhodný integrovaný obvod, který bude realizovat FM rádio, modul rádia by měl obsahovat i možnost příjmu údajů ze systému RDS.
3. Navrhněte obvodové schéma a plošný spoj, tak aby byl obvod kompatibilní s AVR KITem.
4. Ve vývojovém prostředí AVR Studio vytvořte komunikační rutiny pro obsluhu modulu rádia.
5. Vytvořte testovací program pro kontrolu správné funkce modulu.
6. Sepište návody do cvičení popisující práci s vyvinutým modulem.

Seznam doporučené odborné literatury:

[1] Návody cvičení předmětu Architektury počítačů a paralelních systémů. Dostupné na WWW <http://poli.cs.vsb.cz/edu/apps/lab/apps-cvic.pdf>
[2] Dokumentace procesoru ATMEGA32 <http://www.atmel.com/devices/atmega32.aspx>

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. David Seidl, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

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 5. května 2014 Šidel .....

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

V Ostravě 5. května 2014 Šidel .....

Rád bych touto formou poděkoval panu doktoru Davidu Seidlovi za vedení mé bakalářské práce.

Abstrakt

Cílem této práce je návrh a výroba výukového modulu, který umožnuje příjem rádia. Tento modul bude využíván v hodinách předmětu Architektura počítačů a paralelních systémů. Součástí práce je také vytvořit komunikační rutiny pro ovládání modulu a sepsat návody do cvičení pro studenty.

Klíčová slova: Rádio, FM, RDS, Výukový modul

Abstract

Purpose of this thesis is to design and create training module which allows to receive radio. This module will be used in Computer and parallel system architecture classes. This thesis includes implementation of communication routines for module control and create seminar instructions for students.

Keywords: Radio, FM, RDS, Training module

Seznam použitých zkratek a symbolů

AID	–	Application Identification code
AM	–	amplitudová modulace
DPS	–	deska plošných spojů
FM	–	frekvenční modulace
GPIO	–	General-purpose input/output
LTN	–	Location Table Number
ODA	–	Open Data Application
PI	–	Program Identification Codes
PTY	–	Program Type Code
RBDS	–	Radio Broadcast Data System
RCLK	–	Reference clock
RDS	–	Radio Data System
RSSI	–	Recived Signal Strength Indicator
SCL	–	Serial clock
SDA	–	Serial data
SEN	–	Serial enabled
TMC	–	Traffic Message Channel
TP	–	Traffic Program

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Cílem této práce je navrhnout a vytvořit výukový modul pro vývojový KIT využívaný v předmětu Architektura počítačů a paralelních systémů. Modul měl obsahovat digitálně řízený obvod FM rádia. Dále vytvořit komunikační rutiny pro obsluhu modulu. Součástí bakalářské práce je také vytvořit návody do cvičení. Hlavním cílem je umožnit studentům osvojit si ovládání periferního zařízení. Tato práce vyžadovala znalosti z oblastí elektroniky, programování a telekomunikace.

2 Popis AVR-KITu

Výukový AVR-KIT se používá v předmětu Architektura počítačů a paralelních systémů. AVR-KIT je deska plošných spojů, která je osazena procesorem ATmega32 a 2 paticemi ZIF, každá připojena na jeden port procesoru. Dále 2 LED a 4 tlačítka, připojeny na port D procesoru. Nakonec tlačítko reset, které je připojeno na pin $\overline{RESET}$. Na desce jsou umístěny pomocné obvody sloužící k programování procesoru a komunikaci s PC. AVR-KIT je napájen 5V pomocí USB konektoru.[1]

2.1 ATmega32

Kompletní dokumentace k procesoru je ke stažení na stránkách výrobce [3]. Napájecí napětí procesoru je 2,7-5,5V. Vyrábí se v pouzdrech PDIP a TQFP/MLF. ATmega32 obsahuje 32kB programové paměti typu flash, 1024B paměti EEPROM, 2kB datové paměti SRAM. Zdrojem hodinového signálu je jako výchozí nastaven vnitřní RC oscilátor 1MHz. Dalšími možnostmi je přivést vnější hodinový signál, připojit vnější krystal, vnější keramický rezonátor, vnější RC oscilátor nebo změnit nastavení vnitřního RC oscilátoru. ATmega32 obsahuje 4 vstupně výstupní porty, které obsahují navíc alternativní funkce:

- Port A-slouží jako A/D převodník.
- Port B-SPI komunikace, analogový komparátor, čítač/časovač
- Port C-I2C komunikace, JTAG, oscilátor/časovač
- Port D-USART, zdroj vnějšího přerušení, čítač/časovač

3 Rádiové vysílání

Rádiové vysílání se přenáší jako elektromagnetické vlnění. Pro přenos se zvukový signál namoduluje na nosnou vlnu. Nosná vlna má obvykle mnohem vyšší frekvenci než modulační signál.

3.1 Amplitudová modulace

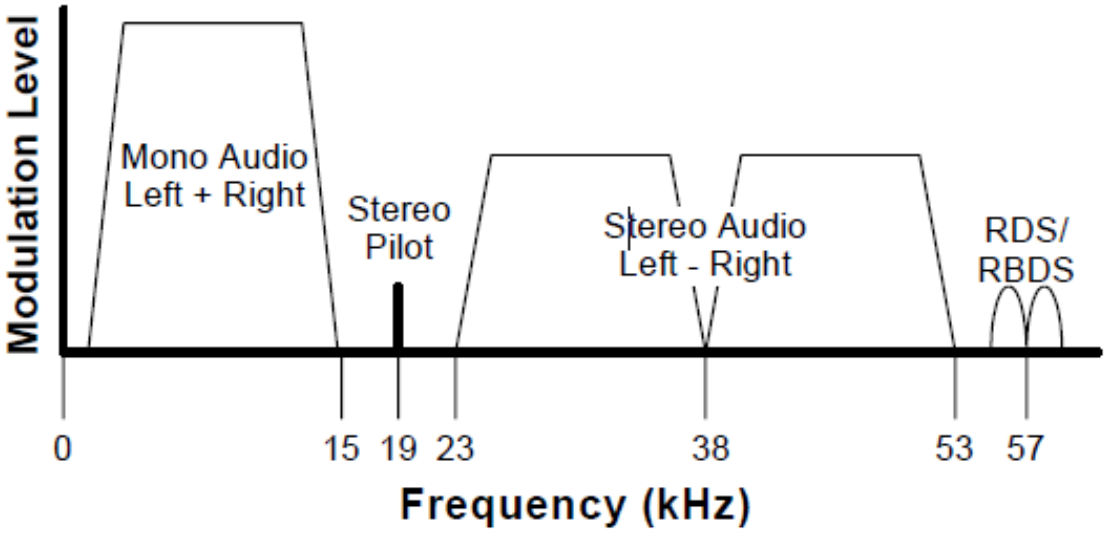
Je to modulační technika používána pro přenos informací na nosné vlně rádiového signálu. Princip modulace je změna amplitudy nosné vlny úměrně vstupnímu signálu. Vysílání AM rádia je nejčastěji na středních vlnách.

3.2 Frekvenční modulace

Frekvenční modulace je typ, modulace při které se mění frekvence nosné vlny v závislosti na vstupním signálu.

3.3 Popis RDS

RDS je doprovodný signál k vysílání FM. Protože je to pouze doprovodný signál, nesmí nikdy narušit hlavní audio signál. Proto je RDS rozšifrovatelné pouze při silném signálu.

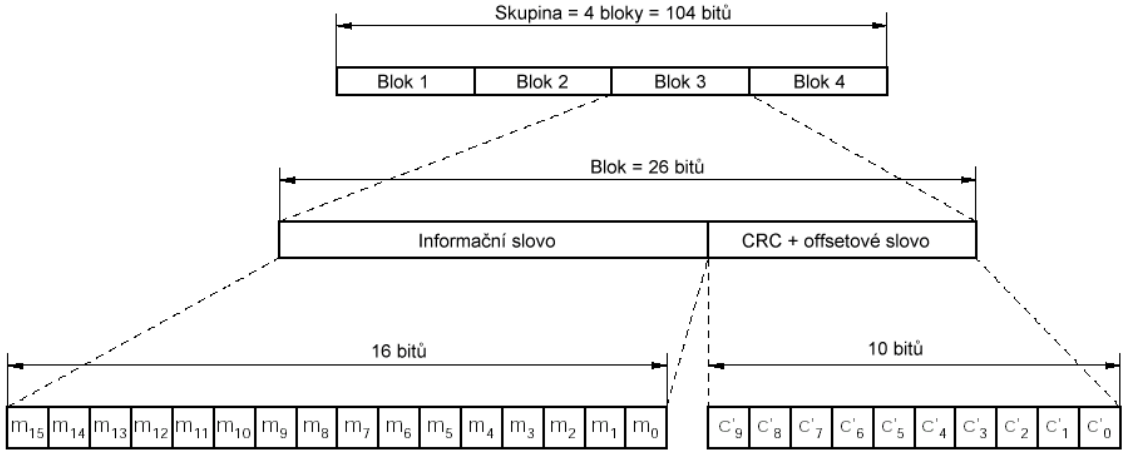


Obrázek 1: Spektrum vysílacího pásma FM [4]

Na obrázku 1 je vidět, že nejnižší frekvenci a nejsilnější signál má vysílání mono audio. To znamená, že přijímač bude nejdříve přehrávat mono, dokud nebude FM signál dostatečně silný pro demodulaci signálu stereo. RDS má nejužší šířku pásma, proto je signál RDS pro přijímač nejtěžší pro dekódování, pokud je signál slabý.

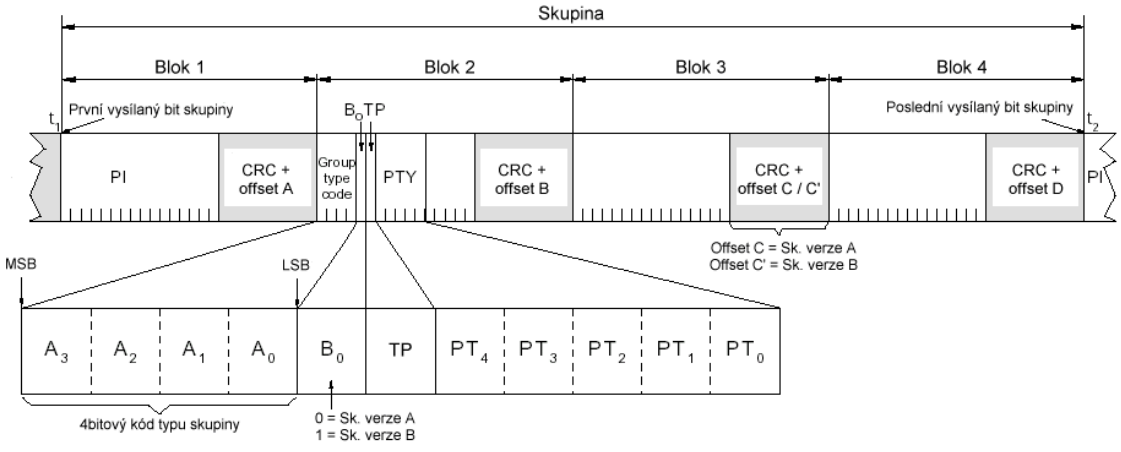
RDS v Evropě a RBDS v severní Americe jsou identické na fyzické vrstvě a velmi podobné na datové a prezentační vrstvě.

3.3.0.1 Struktura dat RDS Na obrázku 2 je vidět, že RDS je přenášeno ve formě 4 datových bloků, každý má 26 bitů obsahu a informace k opravě chyb. Tyto 4 bloky jsou označeny jako skupina. Informace týkající se toho, jaké data skupina obsahuje jsou uloženy v samotných blocích.



Obrázek 2: Struktura dat RDS [7]

Existuje 32 možných typů skupin od běžných až po nouzové. Tabulka je obsažena v příloze9. Na obrázku 3 je vidět strukturu bloků RDS. Informace zakódované v každé RDS skupině má společnou pevně danou strukturu a to podle typu skupiny. Struktura udává formát dat a interval opakování.



Obrázek 3: Struktura bloků RDS [7]

Popis bloků:

Blok 1: První blok každé RDS skupiny obsahuje 16bitový PI kód. Je to identifikační číslo jedinečné pro danou rozhlasovou stanici. PI kód rozhlasové stanice na různých vysílačích je vždy identický.Tato informace spolu se seznamem alternativních vysílacích frekvencí ve skupině 0A, může být použita pro automatické přelaďování.

Blok 2: Druhý blok obsahuje kód typu skupiny, kód verze, TP, PTY a 5 nezařazených bitů.

Kód typu skupiny obsahuje 4 bity a určuje, kterou skupinu datové bloky obsahují. Skupiny RDS jsou označeny čísly 0-15 a písmenem A nebo B, proto má každá skupina dvě verze.

Kód verze označuje, jestli je RDS skupina typu A nebo B.

TP poskytuje příznak, zda stanice bude vysílat informace o dopravě.

Blok 3,4: Obsah je různý v závislosti na typu skupiny a kódu verze. Například: Název programu, Radiotext a informace o alternativní frekvenci.

Hlavní funkce	Typ skupiny který obsahuje tyto informace	Vhodný interval opakování [s]
Kód stanice	všechny	11,4
Kód Typu stanice	všechny	11,4
Kód Dopravní stanice	všechny	11,4
Název služby programu	0A,0B	1
Alternativní frekvence	0A	4
Kód Dopravního hlášení	0A,0B,14B,15B	4
Kód dekodéru	0A,0B,15B	1
Kód identifikace hlasu/hudby	0A,0B,15B	4
Zpráva radiotext	2A,2B	0,2
Informace ostatních sítí	14A	až 2

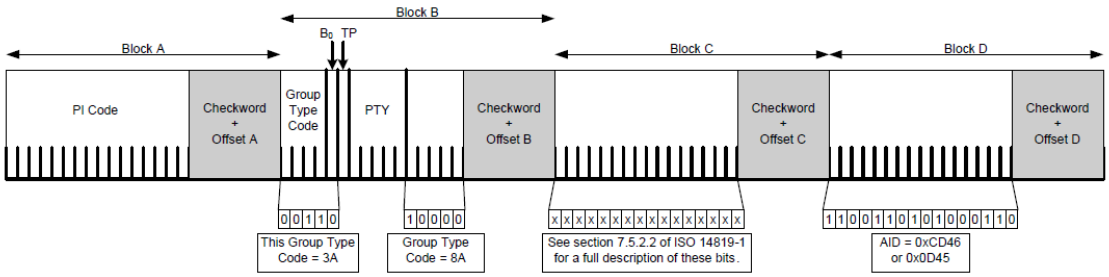
Tabulka 1: Rozmístění funkcí blokových typů a intervaly opakování

Tabulka seznamu typů skupin a jejich popis je v příloze.

3.3.0.2 Alternativní frekvence Je to funkce, která dovoluje přijímačům přepínat mezi dvěmi nebo více frekvencemi stejného vysílání s nejlepším příjmem. Pokud má stanice alternativní frekvence, budou udány v bloku C skupiny typu 0A. Existují 2 různé formáty pro indikaci alternativních frekvencí. Metoda A, která má limit do 25 alternativních frekvencí. Metoda B je využita v situaci, kde není možné se omezit jen na 25 frekvencí. Jelikož funkce alternativní frekvence potřebuje, aby stanice byly relativně blízko, je tato funkce běžnější v Evropě.

3.3.0.3 Kanál dopravních zpráv (TMC) a ODA RDS je schopno sdělit víc než identifikaci stanice, název skladby a aktuální čas. Kromě těchto základních informací je RDS schopno přenášet komplexní informace, jako je stav dopravy a jiné datové proudy. Skupina 3A je klíčová k pochopení, jaké informace jsou odesílány v nespecifických skupinových typech. Skupina 3A obsahuje identifikační kód aplikace (AID) v bloku D. Tento kód indikuje obsah skupiny popsané kódem typu skupiny, nalezených v posledních 5 bitech bloku B.

Na obrázku 4 je vidět skupinu 3A s indikací přítomnosti TMC ve skupině 8A. Když skupina 3A indikuje přítomnost TMC, pak AID kód bude 0xCD46 a kód typu skupiny, indikující která skupina obsahuje RDS informace bude typicky 8A. Je ale dovoleno pro vysílací stanici použít jakoukoliv ODA skupinu. Blok C typu skupiny 3A je různý v závislosti na tom, jakou ODA aplikaci popisuje. Když popisuje TMC aplikaci, nabývá různých významů. Definice těchto bitů je nejdříve určena bity nejvyšší váhy, které se nazývají kód varianty. Pokud je kód varianty nula, pak je zbytek bitů definován jako číslo umístění tabulky (LTN, 6 bitů), indikátor alternativní frekvence (AFI, 1 bit), zpráva geografického měřítka (MGS, 4 bity) a mód přenosu (M, 1 bit). Plné vysvětlení těchto bitů je v normě ISO 14819-1.



Obrázek 4: Struktura skupiny 3A pro indikaci TMC ve skupině 8A [4]

Zdali je TMC informace zašifrovaná je udáno LTN bitem. Pokud je LTN bit nula, pak jsou data zašifrována. Jinak je LTN v kombinaci s kódem země, rozšířeným kódem země a kódem umístění ukazatelem, kterou tabulku umístění použít. Hodnoty LTN pro každou zemi jsou popsány v normě ISO 14819-3.

4 Výběr integrovaného obvodu

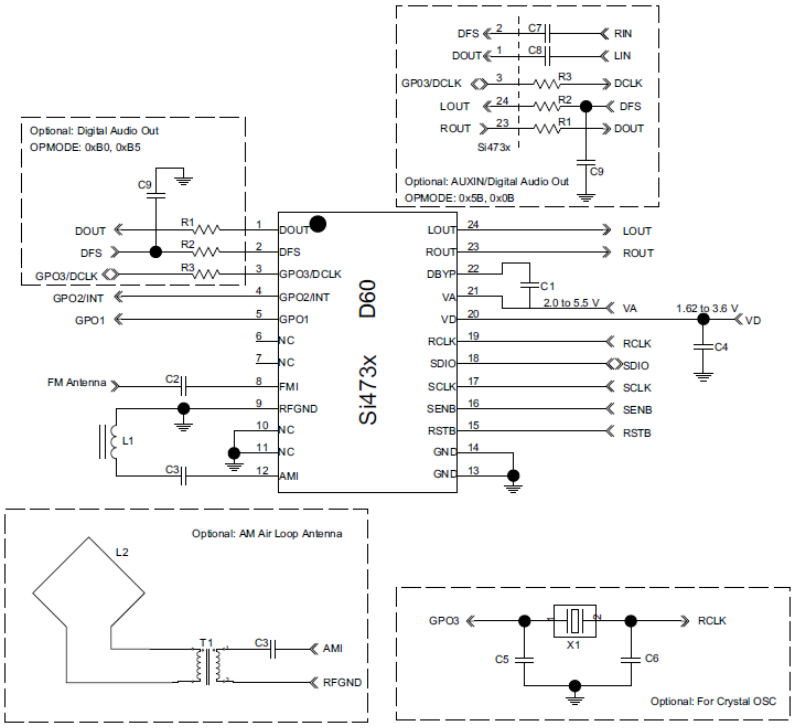
Integrovaný obvod musel mít možnost příjmu FM. Dále musel obsahovat možnost digitálního řízení a musel obsahovat možnost příjmu údajů ze systému RDS.

Rádiový přijímač SI4735 byl zvolen, protože splňoval všechny požadavky a byl dostupnější než jiné typy.

Z ostatních přijímačů splňoval požadované parametry i např. AR1010, ten ale nebyl dostupný a měl pouze variantu pouzdra QFN která nebyla vhodná z důvodu obtížnosti pájení.

4.1 Rádiový přijímač SI4735

Rádiový přijímač řady SI4735 obsahuje funkce příjmu rádiových vysílání AM, FM. Dále příjem RDS, funkce vyhledávání stanic, digitální a analogový zvukový výstup, 2 nebo 3 vodičové řídící rozhraní. Na obrázku 5 je vidět typické zapojení přijímače SI4735.



Obrázek 5: Typické zapojení přijímače SI4735 [6]

Napájení analogové části je 2,7-5,5V, napájení digitální části je ale 1,62-3,6V. AVR-KIT pracuje na napětí 5V, z tohoto důvodu musel být vyřešen problém převodu napěťových úrovní.

Hodinový signál je zajištěn vnitřním oscilátorem, kde je zapotřebí připojení externího krystalu, nebo přivedením externího referenčního signálu.

Přijímač obsahuje GPO výstupy, které můžou být nastaveny do stavů log 0, log 1 a stavu vysoké impedance. GPO výstupy dále obsahují speciální funkce např. GPO2 může poskytovat přerušení a GPO3 slouží pro připojení externího krystalu.

SI4735 obsahuje RDS/RBDS procesor, kde je řešeno dekódování, synchronizace, detekce chyb a oprava chyb. Přijímač obsahuje buffer RDS FIFO, kam se ukládají přijaté RDS skupiny. Příjímač poskytuje nastavitelný příznak přerušení v případech získání RDS synchronizace, ztráty synchronizace a dosažení hranice FIFO.

Funkce vyhledávání vhodných kanálů je implementována na čipu. Přijímač dovoluje nastavit frekvenční rozsah pro funkci vyhledávání kanálů a chování v případě dosažení této hranice. Toto vyhledávání se řídí podle RSSI, kde tato hodnota může být také vyžádána.

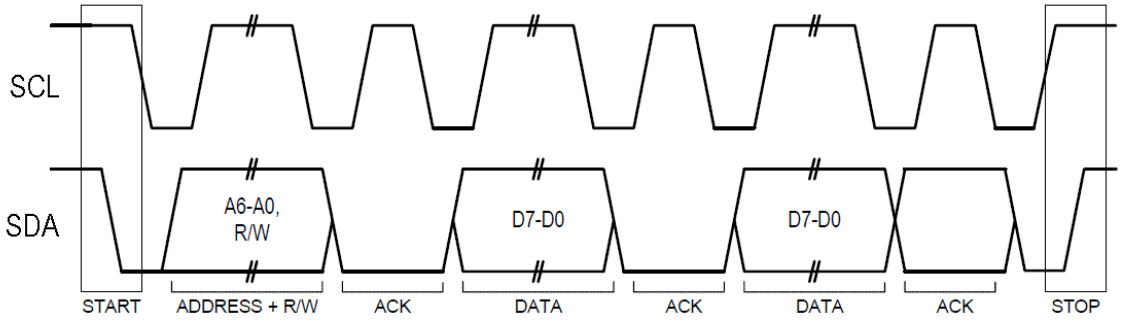
4.1.1 Komunikace

Rádio přijímač SI4735 obsahuje 3 možnosti komunikace:

- 2-vodičové řídící rozhraní (I2C) (výchozí)
- 3-vodičové řídící rozhraní
- SPI komunikace

Typ komunikace je zvolen stavem pinů GPO1 a GPO2 při náběžné hraně $\overline{RST}$.

4.1.1.1 I2C komunikace Průběh komunikace je zobrazen na obrázku 6.Adresa zařízení je nastavena stavem pinu $\overline{SEN}$. I2C používá pro komunikaci piny SDA a SCL.



Obrázek 6: Průběh I2C komunikace [5]

Po inicializaci I2C procesor odešle řídící slovo na pinu SDA. Kontrolní slovo obsahuje adresu zařízení následované příznakem čtení nebo zápisu. Dále odešle bajt příkazu, po kterém může následovat odeslání několika argumentů. Jako argument mohou být odeslány např. adresy registrů. Jakmile jsou všechna data přenesena, procesor komunikaci ukončí.

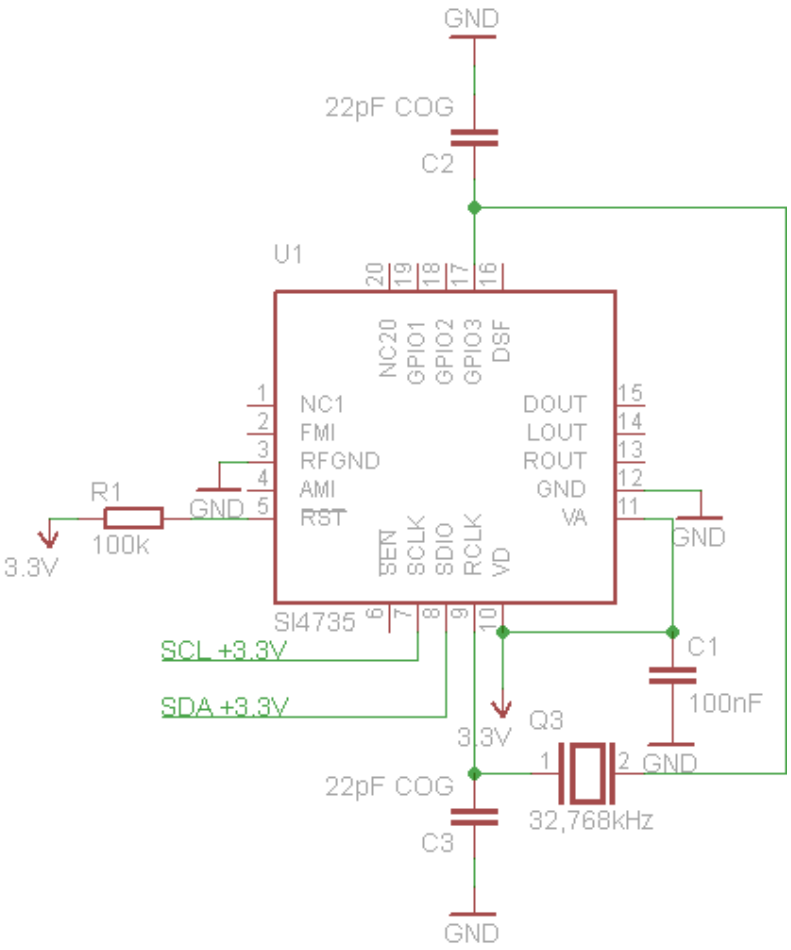
4.1.1.2 3-vodičové řídící rozhraní Pro komunikaci se využívájí piny SDA, SCL a $\overline{SEN}$. Po inicializaci komunikace následuje odeslání 9 bitového kontrolního slova na pinu SDA. Kontrolní slovo obsahuje 3 bity adresy zařízení, bit příznaku zápisu nebo čtení, čtvrtý bit adresy zařízení a 4 bity adresy registru. V případě zápisu následuje odeslání 16bitového slova. V případě čtení přijímač odešle 16bitové slovo.

4.1.1.3 SPI komunikace SPI využívá pro komunikaci piny SDA, SCL a $\overline{SEN}$. Procesor může zvolit příjem dat buďto z SDA nebo GPO1 pinu.Po inicializaci procesor odešle 8bitové kontrolní slovo na SDA pinu. Kontrolní slovo má v případě SPI jednu z 5 hodnot, viz. dokumentace SI4735. V případě zápisu procesor musí odeslat kontrolní slovo a 8 datových bajtů na SDA. V případě čtení procesor musí odeslat kontrolní slovo a číst 1 nebo 16 bajtů.

5 Návrh obvodového schématu a plošného spoje

5.1 Zapojení na nepájivém poli

SI4735 byl nejdříve zapojen do nepájivého pole. Zapojení v této situaci obsahovalo pouze nezbytné součástky pro základní funkci. Obvod byl napájen napětím 3,3V, tedy nebylo zapotřebí stabilizátoru ani převodníku logických úrovní.



Obrázek 7: Schéma zapojení na nepájivém poli

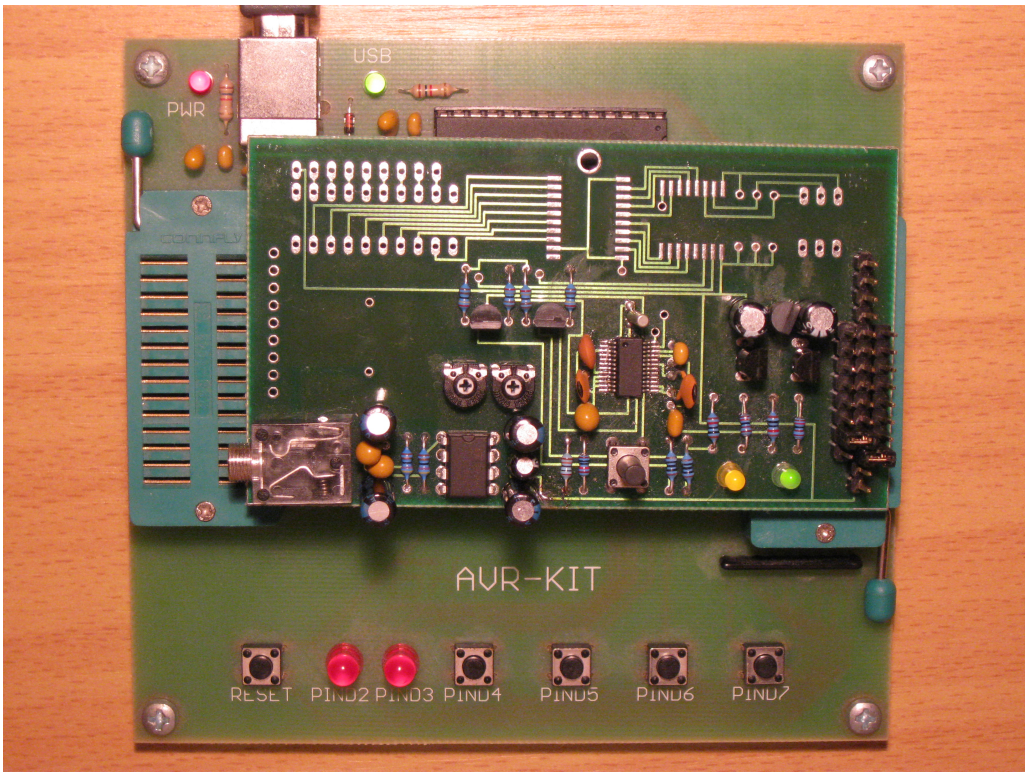
Na obrázku 7 je vidět zapojení přijímače SI4735 v zapojení s minimálním počtem součástek. Pro funkci přijímače je potřebný hodinový signál, ten byl zajištěn připojením krystalu 32,768kH mezi vývody RCLK a GPIO3, který je nezbytný pro funkci vnitřního oscilátoru. Dále napájecí napětí 3,3V přivedené na vývody VD a VA s vyvažovacím kon-

denzátorem 100nF, uzemnění vývodů RFGND a GND. Na vývod $\overline{RST}$ je přes pull-up rezistor 100kΩ přivedeno napětí 3,3V. Obvod komunikoval s procesorem ATMega16. V tomto stavu byla vytvořena a testována I2C komunikace s obvodem SI4735.

5.2 Prototyp

Prototyp modulu rádia už je vyroben jako DPS a obsahuje oproti minimalistickému zapojení stabilizátor napětí na 3,3V a zesilovač. Protože přijímač pracuje na log. úrovni 3,3V a AVR-KIT na 5V, byl přidán převodník logických úrovní. Jako stabilizátor byl zvolen LE33CZ. Reproduktor byl zvolen K 23 PC. Jako zesilovač byl použit TDA2822M a byl zapojen v konfiguraci stereo. Stereo signál byl vyveden do jack konektoru 3,5mm. Do reproduktoru byl přiveden pravý audio výstup. Součástí desky je také obvod expandéru který ale nebyl součástí této práce.

Schéma zapojení je v příloze 15. V tomto stavu byly otestovány veškeré funkce při připojení modulu do AVR-KITu.



Obrázek 8: Prototyp FM modulu zapojen do AVR-KITu

6 Komunikační rutiny

6.1 Inicializace

Po přivedení napětí je přijímač ve stavu PowerDown a v tomto stavu reaguje pouze příkaz PowerUp. Po inicializaci AVR-KITu a I2C sběrnice může být zavolána funkce SI4735_Power_up() který používá příkaz POWER_UP pro nastartování přijímače. V tomto stavu rádiový modul reaguje již na všechny příkazy.

Bit	D7	D6	D5	D4	D3	D2	D1	D0
CMD	0	0	0	0	0	0	0	1
ARG 1	CTSIEN	GPO2OEN	PATCH	XOSCEN	FUNC[3:0]			
ARG 2	OPMODE[7:0]							

Tabulka 2: Argumenty příkazu POWER_UP [5]

Prvním argumentem se určí, zda se má povolit přerušení CTS, GPO2. Dále jestli má přijímač využívat vnitřního krystalového oscilátoru nebo využít vnějšího RCLK signálu. Nakonec jaký typ audio výstupu po přijímači požadujeme. Může se zvolit výstup digitální, analogový nebo oba.

Funkce SI4735_Power_up() nejdříve vybudí I2C linku podmínkou start, poté je možné poslat nejdříve adresu zařízení s příznakem write. Následně se odešle samotný příkaz POWER_UP se dvěmy argumenty.

```
#define POWER_UP 0x01
#define XOSCEN 0x10
#define OPMODE_analog 0x05
void SI4735_Power_up()
{
    I2C_Start();
    I2C_Vystup( adresa_W);
    I2C_Vystup( POWER_UP);
    I2C_Vystup( XOSCEN);
    I2C_Vystup( OPMODE_analog);
    I2C_Stop();
}
```

Výpis 1: Funkce SI4735_Power_up[1] Z výpisu byla vynechána kontrola návratových hodnot

V tomto případě je zvolen analogový výstup a krystalový oscilátor. Komunikace se ukončí podmínkou stop.

6.2 Funkce získání stavu přijímače

Informace ohledně stavu přerušení přijímače je poskytována na prvním místě odpovědi přijímače na určité příkazy, nebo jej lze vyžádat příkazem GET_INT_STATUS.

Bit	D7	D6	D5	D4	D3	D2	D1	D0
STATUS	CTS	ERR	X	X	RQSINT	RDSINT	X	STCINT

Tabulka 3: Odpověd příkazu GET_INT_STATUS [5]

- Bit CTS: Indikuje zda je přijímač připraven na další příkaz.
- Bit ERR: Indikuje chybu.
- Bit RSQINT: Indikuje, zda je změřena kvalita signálu.
- Bit RDSINT: Indikuje, zda jsou připraveny data RDS.
- Bit STCINT: Indikuje, zda je dokončeno ladění.

```
#define GET_INT_STATUS 0x14
int Status()
{
   int x;
   I2C_Start();                    // write
   I2C_Vystup(adresa_W);
   I2C_Vystup(GET_INT_STATUS);
   I2C_Start();                    // read
   I2C_Vystup(adresa_R);
   x=I2C_Vstup();
   I2C_NAck();
   I2C_Stop();
   return x;
}
```

Výpis 2: Funkce SI4735_Int_status [1] Z výpisu byla vynechána kontrola návratových hodnot

6.3 Funkce čtení

Tato funkce implementuje příkaz GET_PROPERTY.

```
#define GET_PROPERTY 0x13
int Get(int registr)
{
    char data[4];
    I2C_Start();                          // write
    I2C_Vystup(adresa_W);
    I2C_Vystup(GET_PROPERTY);
    I2C_Vystup(Null_argument);
    I2C_Vystup(registr >> 8);
    I2C_Vystup(registr);
    I2C_Start();                          // read
    I2C_Vystup(adresa_R);
    for (int x=0;x<3;x++)
    {
        data[x]=I2C_Vstup();
        I2C_Ack();
    }
    data[3]=I2C_Vstup();
    I2C_NAck();
    I2C_Stop();
    return (data[2]<<8)|data[3];
}
```

Výpis 3: Funkce SI4735_Get [1] Z výpisu byla vynechána kontrola návratových hodnot

Funkce Get odesílá adresu zařízení s příznakem write, samotným příkazem GET_PROPERTY, povinným prázdným parametrem a nakonec horní a spodní bajt požadovaného registru. Přijímač odpovídá status bajtem, povinným prázdným bajtem a horní a spodní hodnotou registru.

6.4 Funkce zápisu

Tato funkce implementuje příkaz SET_PROPERTY.

```
#define SET_PROPERTY 0x12
void SI4735_Set(int registr , int  value)
{
   I2C_Start() ;
   I2C_Vystup(adresa_W );
   I2C_Vystup(SET_PROPERTY);
   I2C_Vystup(Null_argument);
   I2C_Vystup(registr  >> 8);
   I2C_Vystup(registr) ;
   I2C_Vystup(value >> 8);
   I2C_Vystup(value);
   I2C_Stop();
}
```

Výpis 4: Funkce SI4735_Set [1] Z výpisu byla vynechána kontrola návratových hodnot

Funkce Set odesílá adresu zařízení s příznakem write, samotným příkazem SER_PROPERTY, povinným prázdným parametrem, horní a spodní bajt zapisovaného registru a horní a spodní bajt hodnoty registru.

6.5 Funkce ladění

Tyto funkce implementují příkaz FM_SEEK_START. Odesílaným argumentem lze určit směr ladění a chování v případě dosažení hranice nastaveného frekvenčního pásma.

```
#define FM_SEEK_START 0x21
#define SEEKUP 0x08
void SI4735_Seek_up()
{
   I2C_Start() ;
   I2C_Vystup(adresa_W );
   I2C_Vystup(FM_SEEK_START);
   I2C_Vystup(SEEKUP);
   I2C_Stop();
}
void SI4735_Seek_down()
```

```
{
    I2C_Start();
    kontrola|=I2C_Vystup(adresa_W);
    kontrola|=I2C_Vystup(FM_SEEK_START);
    kontrola|=I2C_Vystup(Null_argument);
    I2C_Stop();
}
```

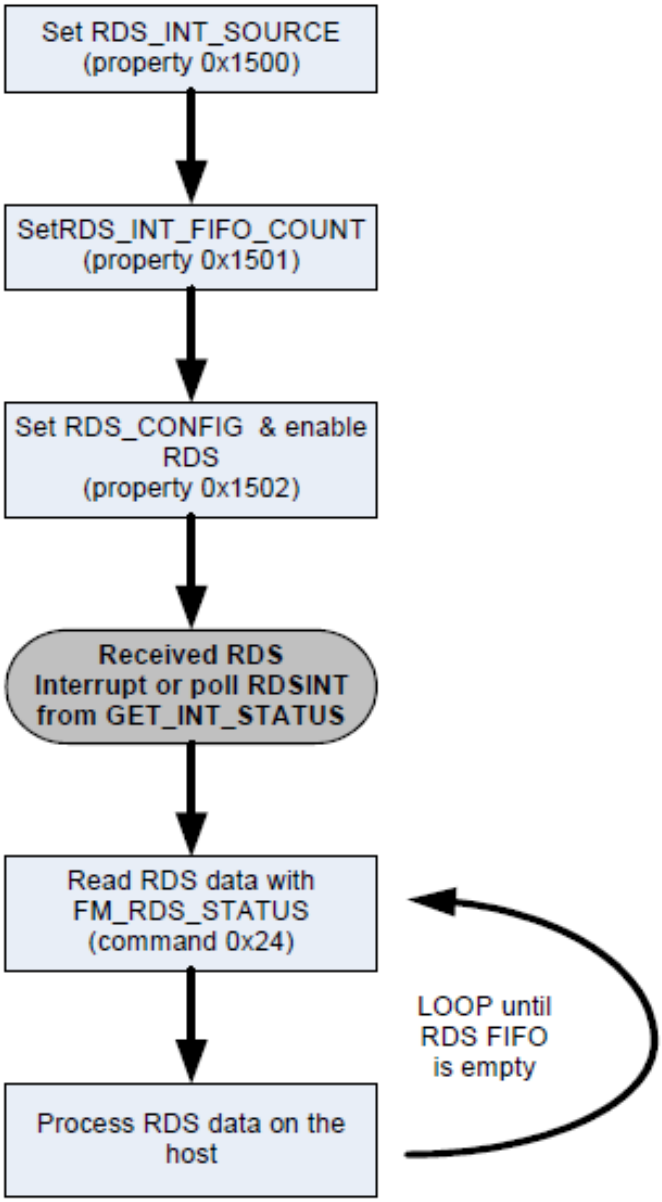
Výpis 5: Funkce ladění [1] Z výpisu byla vynechána kontrola návratových hodnot

Po odeslání adresy s příznakem write a samotným příkazem FM_SEEK_START, je odeslán bajt s volbou směru ladění a zastavení při dosažení hranice nastaveného frekvenčního pásma.

6.6 Funkce získání stavu ladění

Tato funkce implementuje příkaz FM_TUNE_STATUS. Jediným argumentem se nastavuje, zda se má ukončit ladění a jestli vymazat příznak přerušení indikující dokončení ladění. Touto funkcí je možno zjistit, zda se dokončilo ladění, aktuální frekvence, síla signálu a další parametry.

6.7 Rutiny RDS



Obrázek 9: Získávání dat RDS [5]

6.7.1 Inicializace RDS

Po inicializaci AVR-KITu a nastartování přijímače v režimu FM může být provedena inicializace RDS. K inicializaci slouží následující 3 registry.

6.7.1.1 Registr RDS_INT_SOURCE

Registrem se nastavuje přerušení spojené s RDS.

Bit	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Name	0	0	0	0	0	0	0	0	0	0	RDSNEW-BLOCKB	RDSNEW-BLOCKA	0	RDSSYNC-FOUND	RDSSYN-CLOST	RDSRECV

Tabulka 4: Registr RDS_INT_SOURCE [5]

- Bit RDSNEWBLOCKB: Pokud je bit nastaven, vytváří přerušení, když jsou data bloku B nalezeny nebo změněny.
- Bit RDSNEWBLOCKA: Pokud je bit nastaven, vytváří přerušení, když jsou data bloku A nalezeny nebo změněny.
- Bit RDSSYNCFOUND: Pokud je bit nastaven, vytváří přerušení, RDSINT když získá synchronizaci RDS.
- Bit RDSSYNCLOST: Pokud je bit nastaven, vytváří přerušení, RDSINT když ztratí synchronizaci RDS.
- Bit RDSRECV: Pokud je bit nastaven, vytváří přerušení RDSINT když má RDS FIFO vstupů odpovídající hodnotě alespoň RDS_INT_FIFO_COUNT.

6.7.1.2 Registr RDS_INT_FIFO_COUNT

Registrem se nastavuje minimální počet RDS skupin uložených v RDS FIFO předtím než je nastaven RDSRECV. Maximální počet RDS skupin je 25.

Bit	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Name	0	0	0	0	0	0	0	0	RDSFIFOCNT[7:0]							

Tabulka 5: Registr RDS_INT_FIFO_COUNT [5]

- Bit RDSFIFOCNT: Minimální počet RDS skupin uložených v RDS FIFO předtím, než je nastaven RDSRECV.

6.7.1.3 Registr RDS_CONFIG Registrem se nastavuje práh chybovosti a povoluje zpracování RDS. Po přijetí RDS skupiny všechny blokové chyby musí být menší nebo rovno nastaveným hodnotám.

Bit	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Name	BLETHA[1:0]		BLETHB[1:0]		BLETHC[1:0]		BLETHD[1:0]		0	0	0	0	0	0	0	RDSEN

Tabulka 6: Registr RDS_CONFIG [5]

- Bity BLETH(A-D) nastavují práh chybovosti.
- Bit RDSEN povoluje zpracování RDS.

Data se z přijímače získávají příkazem FM_RDS_STATUS. Funkce SI4735_RDS_get() implementuje tento příkaz. Jediný argument tohoto příkazu určuje, zda se mají číst nejstarší nebo nejnovější data, dále možnost vymazat přijímací buffer RDS FIFO a nakonec možnost vymazání příznaku přerušení RDSINT. Přijímač na tento příkaz odpovídá status bajtem a dalšími 12 bajty.

Bit	D7	D6	D5	D4	D3	D2	D1	D0
STATUS	CTS	ERR	X	X	RSQINT	RDSINT	X	STCINT
RESP1	X	X	RDSNEWBLOCKB	RDSNEWBLOCKA	X	RDSSYNCFOUND	RDSSYNCLOST	RDSRECV
RESP2	X	X	X	X	X	GRPLOST	X	RDSSYNC
RESP3	RDSFIFOUSED[7:0]							
RESP4	BLOCKA[15:8]							
RESP5	BLOCKA[7:0]							
RESP6	BLOCKB[15:8]							
RESP7	BLOCKB[7:0]							
RESP8	BLOCKC[15:8]							
RESP9	BLOCKC[7:0]							
RESP10	BLOCKD[15:8]							
RESP11	BLOCKD[7:0]							
RESP12	BLEA[1:0]		BLEB[1:0]		BLEC[1:0]		BLED[1:0]	

Tabulka 7: Odpověd na příkaz FM_RDS_STATUS [5]

- V prvním bajtu jsou informace ohledně synchronizace a přijetí RDS dat.
- V druhém je indikace aktiální synchronizace a indikace přetečení bufferu RDS FIFO
- Třetí udává množství RDS skupin zbývajících v bufferu RDS FIFO.
- 4-5 bajt obsahuje RDS blok A.
- 6-7 bajt obsahuje RDS blok B.
- 8-9 bajt obsahuje RDS blok C.
- 10-11 bajt obsahuje RDS blok D.
- Poslední bajt informuje o stavu přijatých bloků, zda v nich nastala chyba a jestli byla opravena.

Funkce odesílá příkaz FM_RDS_STATUS a v argumentu nastavuje informaci o tom, že chce smazat příznak přerušení. V přijímaných datech ukládá horní a dolní část bloku 2. Poté uloží bloky 3 a 4. Nakonec uloží poslední bajt, který nese informace o chybách v blocích a ukončí komunikaci. Následně ponechá horní 4 bity bloku 2, která určuje typ skupiny. Pokud se jedná o skupinu 2 a nedošlo k chybám, ponechá poslední 4 bity bloku 2 která určuje pozici znaků. Poté uloží všechny znaky do připraveného pole na jejich pozice dané posledními 4 bity bloku 2. Nakonec zadá příznak, že relevantní data byla zapsána do pole.

```
#define FM_RDS_STATUS 0x24
#define INTACK 0x01
void SI4735_RDS_get()
{
    char data[13];
    int  typ;
// write
    I2C_Start();
    I2C_Vystup(adresa_W);
    I2C_Vystup(FM_RDS_STATUS);
    I2C_Vystup(INTACK);
// read
    I2C_Start();
```

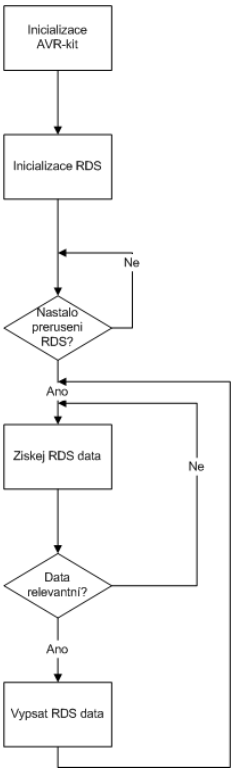
```
    I2C_Vystup(adresa_R );
    for (int x=0;x<12;x++)
    {
        data[x]=I2C_Vstup();
        I2C_Ack();
    }
    data[12]=I2C_Vstup();
    I2C_NAck();
    I2C_Stop();

    if ((data[6]>>4)==2 & data[12]==0) //typ skupiny je 2 a zadne chyby
    {
        pointer=(data[7]&0b00001111)*4;
        for(int x=0;x<4;x++)
        {
            pointer++;
            text[pointer]=data[8+x];
        }
        RDS_povoleni=1;
    }
}
```

Výpis 6: Funkce SI4735_RDS_get [1] Z výpisu byla vynechána kontrola návratových hodnot

7 Testovací program

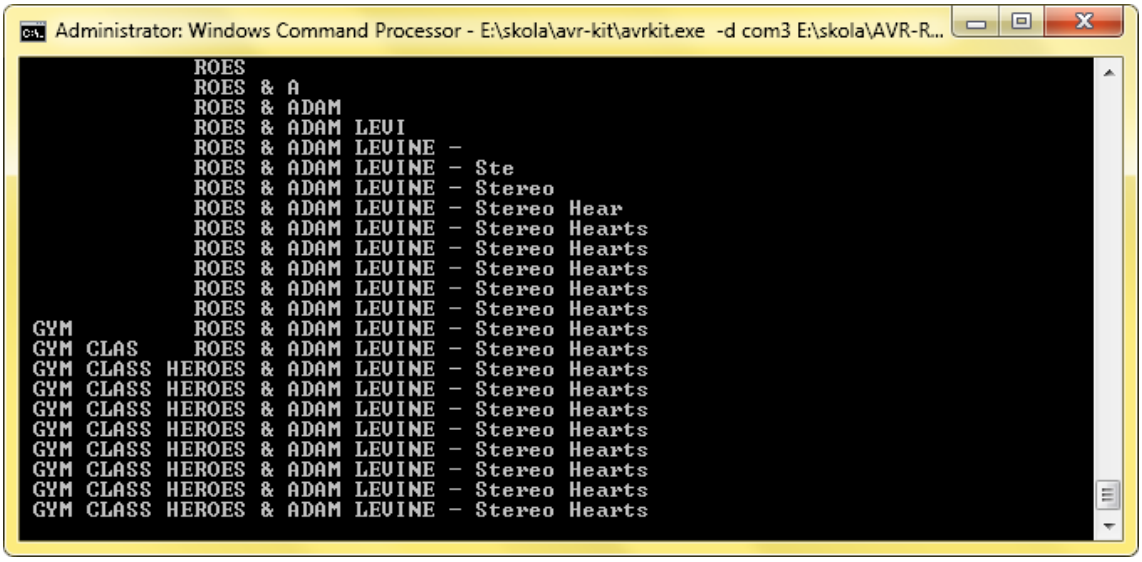
Pro uložení RDS dat je vytvořeno pole char velikosti 64B. Na obrázku 10 je zobrazen průběh čtení dat RDS v hlavní smyčce. Po inicializaci AVR-KITu je spuštěna hlavní smyčka.



Obrázek 10: Cyklus čtení dat RDS

Po provedení inicializace RDS se trvale čte stav přijímače funkcí Status. V případě, že stav přijímače obsahuje indikaci, že jsou data RDS připraveny, povolí se jejich čtení v hlavní smyčce. Čtení se provádí příkazem FM_RDS_STATUS. Tento příkaz je implementován funkcí GET_RDS().

Přečtená data jsou ukládána do jejich příslušných pozic v poli. Pokud byly přečteny relevantní data, je pole vypsáno do konzole. Tímto je zajištena aktuálnost RDS dat. Po změně stanice je pole smazáno, zastavena smyčka čtení RDS dat a znovu se čeká na příznak připravenosti dat RDS.



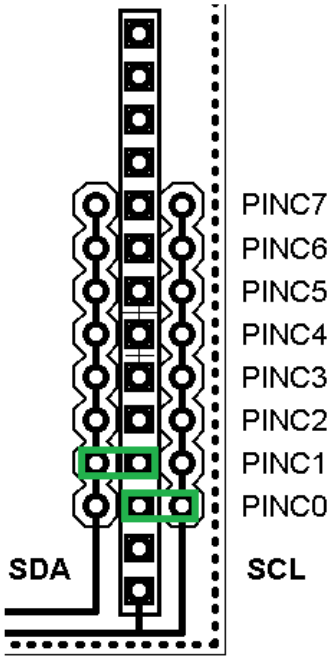
Obrázek 11: Výpis radiotextu do konzole

Z obrázku je vidět, jak jsou přijímána data RDS. Procesor v této ukázce vypisuje do konzole pouze data obsahující radiotext. V každé přijaté skupině jsou 4 znaky. Přijímač v tomto případě nejdříve zachytil 4.skupinu a každou další skupinu kterou vysílač odeslal, do celkového počtu 64 znaků. Poté začali data přicházet od začátku.

8 Návod do cvičení

8.1 FM modul

Hlavní součástí FM modulu je rádiový přijímač SI4735. Kompletní dokumentace je dostupná na stránce výrobce [6]. Dále je připravena zkrácená verze dokumentace. Modul obsahuje samotný rádiový přijímač, zesilovač, reproduktor, jack konektor, 2 LED pro indikaci komunikace a tlačítko. Audio výstup je vyveden do jack konektoru, kde je možné připojit sluchátka nebo po dobu stisknutí tlačítka je aktivní reproduktor. Modul je připojen na port C procesoru. S modulem se komunikuje pomocí I2C, kde se nejdříve zvolí piny SCL a SDA připojením jumperů mezi prostředním a krajním kolíkem viz obr12. Jako výchozí nastavení je použit PINC0 jako SCL a PINC1 jako SDA. Pokud jsou kolíky připojeny jinam, je nutno změnit definici SCL_PIN a SDA_PIN v souboru i2c.c na jejich příslušné piny.



Obrázek 12: Pozice pinů na modulu

8.1.1 Demo program pro FM modul

Program demo je obsažen v souboru Radio.hex. Inicializace nastaví kromě jiného příjem na FM. FM modul je ovládán pomocí tlačítek AVR-KITu

- PIND4: vyhledávání dolů
- PIND5: vyhledávání nahoru
- PIND6: Hlasitost dolů
- PIND7: Hlasitost nahoru

V případě, že modul nalezne vhodnou frekvenci a tato stanice vysílá vhodné RDS data jsou poté vypsána do konzole společně s indikací síly signálu hodnotou RSSI(Received signal strength indication) v jednotkách dBμV.

8.1.2 Vyžádání stavu přijímače

Pro komunikaci s FM modulem se nejdříve odešle bajt adresy zařízení s příznakem zápisu nebo čtení, následován bajtem samotného příkazu a případnými bajty argumentů. Adresa zařízení je 0x22 pro zápis a 0x23 pro čtení.

Po úspěšné inicializaci se může vyžádat status FM modulu příkazem GET_INT_STATUS 0x14. Tento příkaz nemá žádné další argumenty.

Bit	D7	D6	D5	D4	D3	D2	D1	D0
STATUS	CTS	ERR	X	X	RSQINT	RDSINT	X	STCINT

Tabulka 8: Odpověd status bajtu [5]

V tabulce 8 je vidět, že odpověď modulu je jediný bajt.

- CTS (Clear to send) - indikuje, že zařízení je připraveno na další příkaz.
- ERR (Error) - indikace chyby.
- RSQINT (Received Signal Quality Interrupt) - indikuje, že zařízení změřilo kvalitu signálu

- RDSINT (RDS interrupt) - indikuje, že zařízení má připravena data RDS.
- STCINT (Seek/Tune interrupt komplete) - indikuje, že zařízení dokončilo vyhledávání.

8.1.3 Čtení a zápis

Registry mají 16b adresu i hodnoty, proto se musí číst horní a spodní část a následně tyto data sloučit. Čtení se provádí příkazem GET_PROPERTY 0x13 a zápis SET_PROPERTY 0x14.

Při čtení registru se odesílá samotný bajt příkazu, povinný prázdný bajt, horní a spodní bajt adresy registru. Ke zjištění správnosti čtení lze přečtené hodnoty porovnat s výchozími hodnotami.

Pro zápis registru se odesílá samotný bajt příkazu, povinný prázdný bajt, horní a spodní bajt adresy registru, horní a spodní bajt hodnoty, na kterou se má registr nastavit.

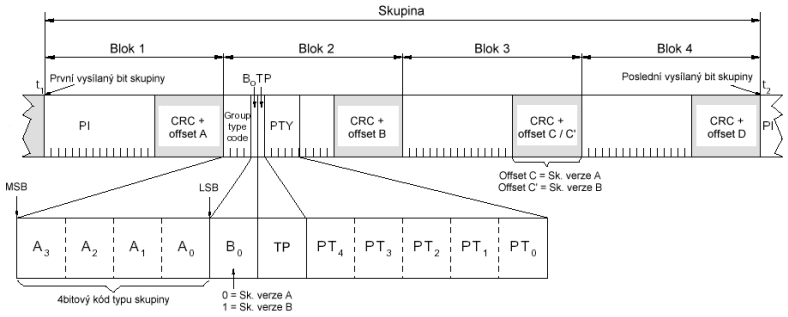
8.1.4 Ladění

Naladit stanici lze přímo příkazem FM_TUNE_FREQ 0x20, nebo vyhledáním vhodné stanice příkazem FM_SEEK_START 0x21. V případě příkazu FM_TUNE_FREQ v prvním a posledním argumentu může být ponechána 0. Argument 2 a 3 udává horní a spodní bajt hodnoty frekvence, která se má naladit.

V případě vyhledávání stanice příkazem FM_SEEK_START v jediném argumentu se nastavuje směr vyhledávání a WRAP, který udává jestli se vyhledávání při dosažení hranice zastaví nebo se vrátí na poslední známou frekvenci.

8.1.5 RDS

RDS jsou doplňkové informace k vysílání FM. Jeden z těchto doplňkových informací je radiotext, který obsahuje informace např. název stanice nebo název přehrávané skladby. RDS data jsou odesílány ve skupinách obr.13.



Obrázek 13: Pořadí bloků RDS [7]

- Blok 1: obsahuje Identifikační kód stanice.
- Blok 2: obsahuje kód typu skupiny který označuje jaké data jsou přijaty.
- Bloky 3 a 4 obsah závisí na typu skupiny.

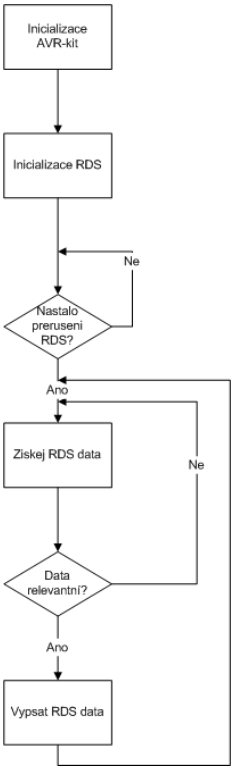
RDS data se ukládají do RDS bufferu FIFO na přijímači.

8.1.5.1 Inicializace RDS

Pro inicializaci RDS je zapotřebí nastavit 3 registry.

- RDS_INT_SOURCE 0X1500: registr nastavuje možnosti RDS přerušení. Doporučená hodnota je 1.
- RDS_INT_FIFO_COUNT 0X1501: registr nastavuje minimální množství skupin uložených v bufferu RDS FIFO než dojde k přerušení. Doporučená hodnota je 4.
- RDS_CONFIG 0X1502: registr povoluje RDS a nastavuje hranice chybovosti, která určuje jestli se daná skupina uloží nebo ne. Doporučená hodnota je 0XAA01.

8.1.5.2 Čtení dat RDS Na obrázku 14 je ukázáno, jak by se měly data RDS číst.



Obrázek 14: Cyklus čtení dat RDS

Jestli jsou data připravena se zjistí přečtením příznaku přerušení ve status bajtu přijímače.

V případě, že by se data z FIFO četly pomaleji, než se naplňuje, dojde k nekonzistenci dat, proto by se měla snížit hodnota funkce I2C_delay v souboru i2c.c na hodnotu např. 10μs.

Pro čtení těchto dat se používá příkaz FM_RDS_STATUS 0x24. S příkazem se odesílá jediný argument, kde se volí jestli se bude číst nejnovější nebo nejstarší data, možnost vymazání FIFO, možnost vymazání příznaku přerušení.

V odpovědi je obsaženo nejdříve několik bajtů s informacemi ohledně synchronizace. RDSFIFOUSED udává zbývající počet RDS skupin ve FIFO. Počínaje pátým bajtem začínají přicházet samotná data skupiny RDS. Nakonec je odeslán bajt s informacemi o případných chybách.

Radiotext má kód skupiny 2 a délku vždy 64 znaků. V případě radiotextu bloky 3 a 4 obsahují každý 2 znaky. RDS se začne přijímat v náhodnou chvíli a nemusí žačít číst od prvního znaku. K určení pozice znaku slouží poslední 4 bity bloku 2. Když je tato hodnota 0, pak jsou znaky radiotextu na pozici 0,1,2,3. Pokud je hodnota 1, tak jsou znaky na pozici 4,5,6,7 atd.

Název programu (PS) má kód skupiny 0 a délku 8 znaků. Dvojice znaků je uložena v bloku 4. Pozice znaků je dána posledními 2 bity bloku 2.

9 Závěr

Na začátku bylo třeba nastudovat problematiku mikroprocesorů a rádia. Nejrozsáhlejší čast byla příjem dat RDS. Dále bylo potřeba zvolit správný integrovaný obvod. Následovalo zapojení na nepájivém poli, kde byla vytvořena a odzkoušena funkčnost komunikace s SI4735. Poté byly vybrány všechny zbývající součástky pro plnou funkci modulu a bylo vytvořeno zapojení na desce plošných spojů. Tato deska plošných spojů byla osazena všemi součástkami rádiového modulu a byla vyzkoušena kompatibilita komunikace s AVR-KITem. Pak byl implementován příjem dat RDS. Nakonec byl vytvořen pro studenty návod do cvičení.

10 Literatura

[1] OLIVKA, Petr a David SEIDL. *Návody do cvičení* [online]. 2011,[cit. 2013-09-03]. Dostupné z: http://poli.cs.vsb.cz/edu/apps/lab/apps-cvic.pdf

[2] OLIVKA Petr. *avr-kit.zip* [online].2013 [cit. 2013-09-03]. Dostupné z: http://poli.cs.vsb.cz/edu/apps/lab/avr-kit.zip

[3] Atmel Corporation. *ATmega32(L)* [online]. 2011, 2503Q–AVR–02/11 [cit. 2013-09-03]. Dostupné z: http://www.atmel.com/Images/doc2503.pdf

[4] SILICON LABORATORIES. *AN243: USING RDS/RBDS WITH THE Si4701/03*[online]. 2007 , Rev. 0.2 3/07[cit. 2013-09-03]. Dostupné z: http://www.silabs.com/Support%20Documents/TechnicalDocs/AN243.pdf

[5] SILICON LABORATORIES. *AN332: SI47XX PROGRAMMING GUIDE*[online]. 2010 , Rev. 0.5 5/10[cit. 2013-09-03]. Dostupné z: http://www.silabs.com/Support%20Documents/TechnicalDocs/AN332.pdf

[6] SILICON LABORATORIES. *Si4730/31/34/35-D60: BROADCAST AM/FM/SW/LW RADIO RECEIVER*[online]. 2011 , Rev. 1.1 11/11[cit. 2013-09-03]. Dostupné z: https://www.silabs.com/Support%20Documents/TechnicalDocs/Si4730-31-34-35-D60.pdf

[7] KOLÁŘ, Jan.*Stručně o systému RDS*[online]. 2012 [cit. 2013-09-03]. Dostupné z: http://www.pira.cz/RDS.pdf

[8] VÁŇA, Vladimír. *Mikrokontroléry ATMEL AVR: popis procesorů a instrukční soubor*. 1. vyd. Praha: BEN - technická literatura, 2003, 335 s. ISBN 80-730-0083-0.

Seznam Příloh

Příloha A: Tabulka RDS skupin bloku 2

Příloha B: Kompletní obvodové schéma

Příloha C: Schéma desky plošných spojů

Příloha D: CD s elektronickými přílohami:

-Radio.c - zdrojový kód programu

-Radio.h - hlavičkový soubor programu

-Radio.hex - zkompilovaný program pro nahrání do AVR-KITu

-avr-kit.zip - připravené programy pro mikropočítač ve formě zdrojových kódů

-STI0030.pdf - text bakalářské práce

-Návody do cvicení.pdf - návody do cvičení pro studenty

-Kompletní obvodové schéma.pdf - příloha B

-Schéma desky plošných spojů.pdf - příloha C

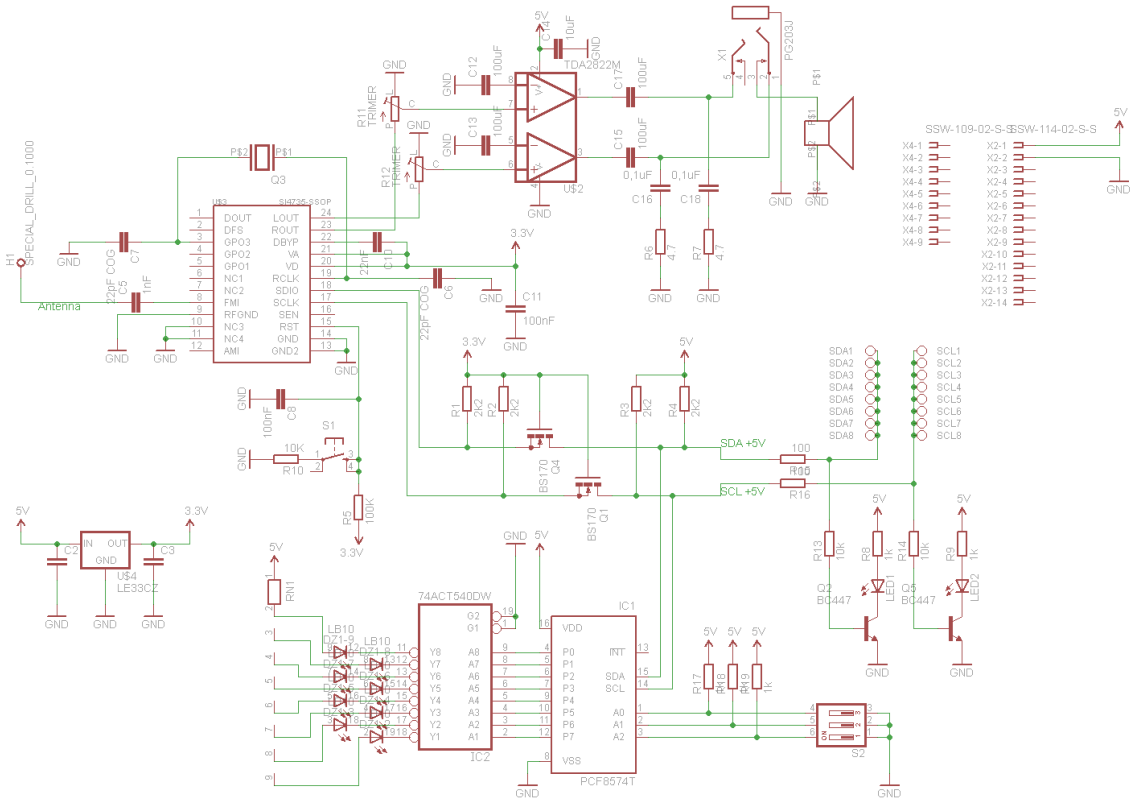
-Si4735 Programming Guide-Zkrácené.pdf - zkrácená dokumentace pro studenty

Příloha A

Group Type	Group Type Code/Version					Flagged in Type 1A Groups	Description
	A_3	A_2	A_1	A_0	B_0		
0A	0	0	0	0	0		Basic Tuning and Switching Information only
0B	0	0	0	0	1		Basic Tuning and Switching Information only
1A	0	0	0	1	0		Program Item Number and Slow Labeling Codes only
1B	0	0	0	1	1		Program Item Number
2A	0	0	1	0	0		Radio Text only
2B	0	0	1	0	1		Radio Text only
3A	0	0	1	1	0		Applications Identification for ODA only
3B	0	0	1	1	1		Open Data Applications
4A	0	1	0	0	0		Clock Time and Date only
4B	0	1	0	0	1		Open Data Applications
5A	0	1	0	1	0		Transparent Data Channels (32 channels) or ODA
5B	0	1	0	1	1		Transparent Data Channels (32 channels) or ODA
6A	0	1	1	0	0		In-House Applications or ODA
6B	0	1	1	0	1		In-House Applications or ODA
7A	0	1	1	1	0	Y	Radio Paging or ODA
7B	0	1	1	1	1		Open Data Applications
8A	1	0	0	0	0	Y	Traffic Message Channel or ODA
8B	1	0	0	0	1		Open Data Applications
9A	1	0	0	1	0	Y	Emergency Warning System or ODA
9B	1	0	0	1	1		Open Data Applications
10A	1	0	1	0	0		Program Type Name
10B	1	0	1	0	1		Open Data Applications
11A	1	0	1	1	0		Open Data Applications
11B	1	0	1	1	1		Open Data Applications
12A	1	1	0	0	0		Open Data Applications
12B	1	1	0	0	1		Open Data Applications
13A	1	1	0	1	0	Y	Enhanced Radio Paging or ODA
13B	1	1	0	1	1		Open Data Applications
14A	1	1	1	0	0		Enhanced Other Networks Information only
14B	1	1	1	0	1		Enhanced Other Networks Information only
15A	1	1	1	1	0		Defined in RBDS only
15B	1	1	1	1	1		Fast Switching Information only

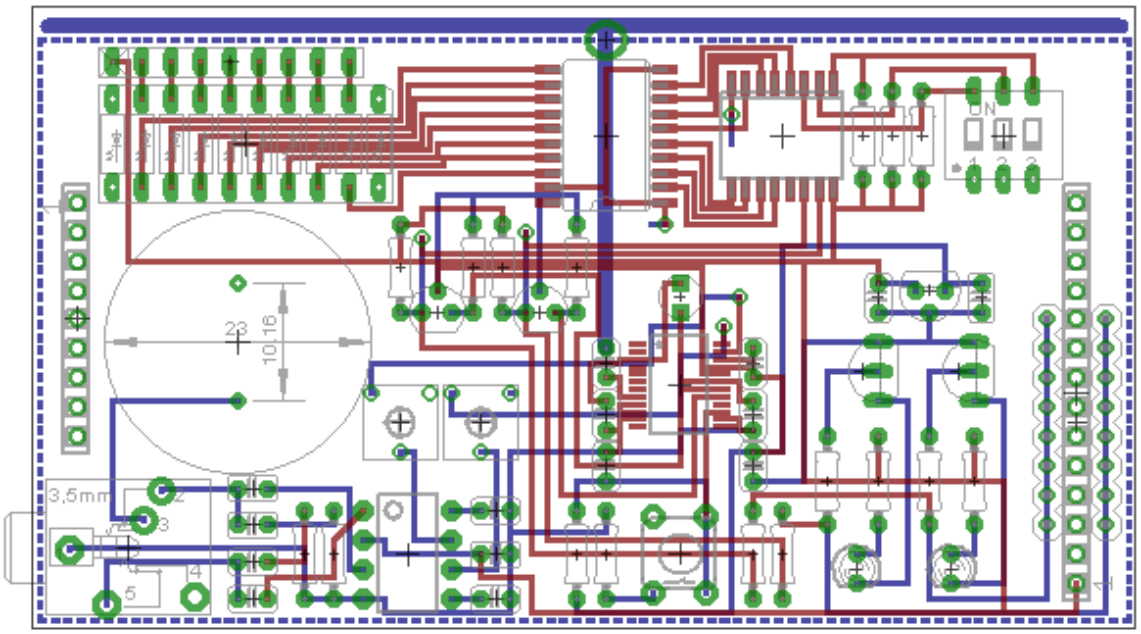
Tabulka 9: Skupiny bloku 2 [4]

Příloha B



Obrázek 15: Kompletní obvodové schéma

Příloha C



Obrázek 16: Schéma desky plošných spojů

Příloha D

Demo program Radio.hex může být nahrán do AVR-KITu pomocí avrkit.exe, který je k dispozici v přiloženém archívu avr-kit.zip nebo ke stažení ze stránek http://poli.cs.vsb.cz/edu/apps/lab/avr-kit.zip, postup nahrávání programu do AVR-KITu je k dispozici na stránkách http://poli.cs.vsb.cz/edu/apps/lab/apps-cvic.pdf.

Samotný zdrojový kód potřebuje ke kompilaci soubory i2c.c , avrkit.c a jejich hlavičkové soubory, které jsou v přiloženém archívu avr-kit.zip nebo ke stažení ze stránek http://poli.cs.vsb.cz/edu/apps/lab/avr-kit.zip. Nahravání do AVR-KITu opět pomocí avrkit.exe.