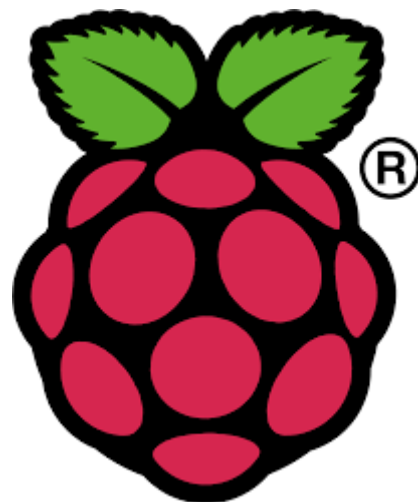


Tuner FM pro Raspberry Pi

„Malina“ pro kterou nebude problém generovat řídicí signály pro I2C. Na internetu najdeme velké množství příkladů jak ovládat I2C.

Popis zapojení:

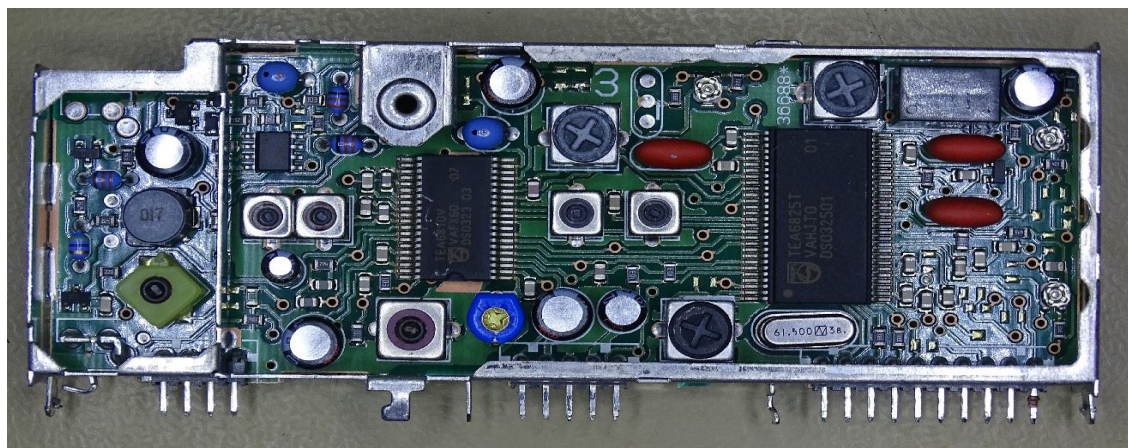
Základem je „profesionální“ tuner, který byl osazován do továrních autorádií. Použijeme hotový tuner bez jakýkoliv úprav, pouze připojíme anténu, napájení a řídicí signály. Připojení tuneru je obdobné jako v autorádiu CDR 2005, kterým byly vybavovány automobily značky Opel. Když se podíváme pod kryty tuneru, zjistíme, že je osazen obvody Philips TEA6810V a TEA6825T. Tato skutečnost je pro nás dostačující, abychom mohli tento tuner řídit po sběrnici I2C. Pro další zpracování využijeme vyvedené nízkofrekvenční signály OUT R a OUT L. Pro možnost SW rozšíření máme vyvedeny na ovládací konektor i signály MPX/RDS a LEVEL.



Parametry tuneru:

Vstupní impedance: 100 Ω
Frekvenční rozsah : 87,5 – 108 MHz FM
531 – 1602 kHz MW Europe (9 kHz grid)
530 – 1620 kHz MW USA (10 kHz grid)
153 – 279 kHz LW
26 dB S/N citlivost: < 10 dB μ V FM
< 27 dB μ V MW
< 33 dB μ V LW
RDS citlivost: < 27 dB μ V
S/N ratio (60 dB μ V): > 57 dB FM mono
> 52 dB AM
a - 3 dB: 15 dB μ V $\pm$ 4 dBFM

uvedené údaje jsou převzaty z parametrů výše citovaného autorádia a měly by být stejné.



Nf signály OUT R a OUT L jsou přes odpory R121 a R124 a kondenzátory C1 a C2 přivedeny na 1. stereo vstup digitálního audio procesoru TDA7313 od firmy ST microelectronics. Jedná se o obvod, který přepíná až 3 stereo vstupy, řídí hlasitost a umožňuje nastavovat hloubky a výšky v obou kanálech. Veškeré řízení tohoto obvodu je po sběrnici I2C. Zapojení je katalogové a je převzato s minimálními změnami z originálního datasheetu tohoto obvodu.

Zapojení ovládacího konektoru X6 – připojení RASPI:

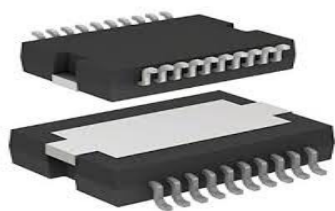
Pin	Signál	Úroveň	Poznámka
1	GND	0 V	Signálová a napájecí zem
2	MPX/RDS	2V šš	Signál pro dekódování RDS
3	EN	Log. H	Spínač napájecího zdroje
4	LEVEL	0- 5V	Úroveň signálu FM
5	ST-BY	Log. L	St-by signál (MUTE) koncového zesilovače
6	SDA	0V / 5V	Data sběrnice I2C
7	+3.3V	napájení	Napájení pro řídicí jednotku s mikroprocesorem (800mA)
8	SCL	0V / 5V	Hodiny sběrnice I2C
9	+5V	napájení	Nespínané pro řídicí jednotku s mikroprocesorem (800mA)
10	+12V	napájení	Pomocné napájení (500mA)

Při prvních pokusech zapojujeme pouze zvýrazněné signály.

Napájení modulu zabezpečuje střídavé napětí 12V přivedené na konektor X5 - POWER. Je možno použít napájení z transformátoru, kde napětí by nemělo přesáhnout 12V a výkon by měl být alespoň 70 W. Vhodné jsou toroidní transformátory určené k napájení halogenových světel, které mají sekundární vinutí 11,5V a bývají vybaveny tepelnou pojistkou.



Lze použít i stejnosměrný napájecí zdroj, který by neměl mít vyšší napětí než 14V a dodat by měl proud alespoň 4A. V tomto případě je možno vynechat usměrňovací můstek a nahradit ho propojkami, ale není to nutné.



Multifunkční napěťový regulátor L5956 od firmy ST nám stabilizuje napětí +8.5V a 5V pro tuner a v 3.3V pro případné další ovládací obvody mimo desku. Je umístěn v pouzdru Power SO20 a umožňuje přiložit chladič v případě, že by se přehříval.

Odkazy na datasheety použitých obvodů:

http://pdf.datasheetcatalog.com/datasheet/philips/TEA6810V_TEA6811V_2.pdf

http://www.nxp.com/documents/data_sheet/TEA6823T_6825T.pdf

<http://www.promelec.ru/pdf/tda7313.pdf>

<http://www.audiosound.info/katlisty/TDA7375.pdf>

<http://pdf1.alldatasheet.com/datasheet-pdf/view/234416/STMICROELECTRONICS/L5956.html>

<http://www.raspi.cz/2012/11/1-wire-snimace-na-i2c-presnejši-mereni-teploty/>

Rozpiska materiálu modulu TUNER_RASPI:

Název	Počet	Hodnota	Pouzdro	Pozice
Keramický kondenzátor	4	100	C0805	C118, C119, C124, C125
Keramický kondenzátor	1	1n5	C0805	C122
Keramický kondenzátor	2	2n7	C0805	C113, C114
Keramický kondenzátor	13	100n	C0805	C100, C101, C104, C105, C106, C107, C108, C109, C116, C117, C120, C121, C123
Keramický kondenzátor	2	330n	C0805	C102, C103
Keramický kondenzátor	4	1u	C0805	C110, C111, C112, C115
Elektrolytický kondenzátor	9	1u	PAN_B	C1, C2, C7, C8, C9, C10, C16, C17, C18
Elektrolytický kondenzátor	9	100u	PAN_D	C3, C4, C11, C12, C13, C14, C15, C20, C21
Elektrolytický kondenzátor	1	100u/25	PAN_F	C19
Elektrolytický kondenzátor	2	3G3	PAN_H	C5, C6
Odpor SMD	6	100	R0805	R100, R101, R114, R115, R116, R117
Odpor SMD	7	1k5	R0805	R107, R108, R110, R111, R120, R121, R124
Odpor SMD	6	1k8	R0805	R103, R105, R109, R119, R122, R123
IO - TDA7313	1	TDA7313	DIL28	IC2
IO - TDA7375	1	TDA7375	MULTIW15	IC1
IO - L5956	1	L5956	PWRSO20	IC3
Tuner FM	1	FM,AM	Spec.	TEA, TEA
Dioda můstek	1	KBL10	4A	Můstek - D2
LED	1	TLXE	G	Indikace PWR - D1
Indukce	1	270u	Toroid	Vinutá L1
Indukce	3	0R	T1206	Feritový beam - L100, L101, L102
Ant	1	-		Anténní konektor
Fuse	1	4A		Pojistka s pouzdem - F1
Konektor	2	MTA3	2.54	Konektor vstupů – IN1, IN2
Konektor	1	5mm	MSTBV2	Napájení - X5
Konektor	4	Faston	6.3	Výstup REPRO - X1, X2, X3, X4
Konektor	1	10H_R	2x2.54	Ovládací konektor - X6
DPS	1	FR4	118x57	Základní DPS – oboustranná prokovená
Chladič	1	Al		Chladič výkonových obvodů

To je vše, co se týká HW. Vše ostatní je již SW záležitost a je jedno, jestli se modul tuneru bude ovládat například vlastní aplikací s mikrokontrolerem PIC či ATMEL nebo se bude ovládat pomocí převodníku USB / I2C, kterých najdeme na internetu více. Primárně byl ale tento modul vyvinut pro aplikace s Raspberry Pi. Jenom poznámka : pull up rezistory pro I2C sběrnici osadíme R1 a R2, protože jsou připojeny na 3.3V napájení jako má „Malina“, rezistory R112 a R113 neosazujeme.

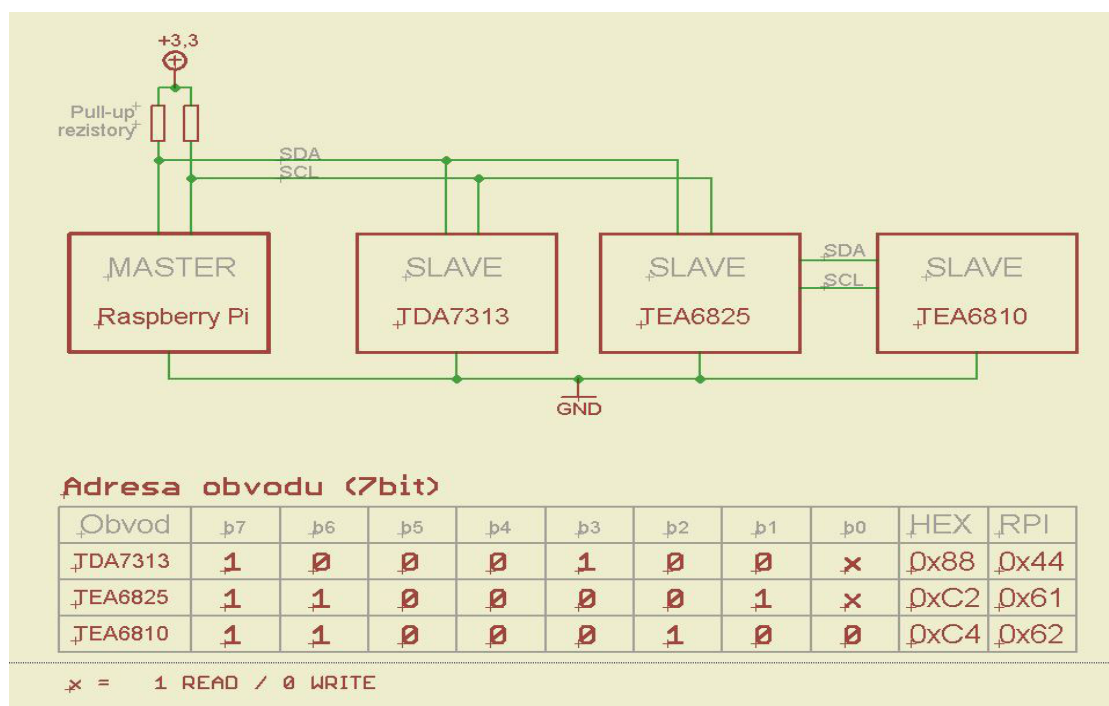
Než začneme s aplikací Raspberry Pi, bude vhodné si něco málo nastudovat jednak o „Malině“, například na českém portále : www.raspi.cz nebo na originálním anglickém : www.raspberrypi.org. Dále bude dobré něco vědět i o sběrnici I2C. Informací je dostupných mnoho, základní charakteristika je na Wikipedii, následné informace byly převzaty z HW serveru:

Stručný popis sběrnice I2C a její praktické využití

I²C bus je zkratka která vznikla z IIC bus, tedy Internal-Integrated-Circuit Bus. Jak již název napovídá, jedná se o interní datovou sběrnici sloužící pro komunikaci a přenos dat mezi jednotlivými integrovanými obvody většinou v rámci jednoho zařízení. Vyvinula ji firma Philips přibližně před 20 lety a od té doby prošla několika vylepšeními, ale o tom až později. V dnešní době tento způsob "komunikace" podporuje řada integrovaných obvodů nejen firmy Philips. Jedná se především o mikrokontroléry, sériové paměti, inteligentní LCD, audio a video obvody, a/d a d/a převodníky a některé další digitálně řízené obvody. Hlavní výhodou je, že obousměrný přenos probíhá pouze po dvou vodičích - "data SDA (serial data)" a "hodiny SCL (serial clock)". To především u mikrokontrolérů výrazně optimalizuje nároky na počet vstupně-výstupních pinů a celkově zjednodušuje výsledné zapojení. Na jednu sběrnici může být připojeno více integrovaných obvodů. V základní verzi jsou obvody adresovány 7bitově a v rozšířené verzi 10bitově. To umožňuje připojení 128 respektive 1024 čipů s různou adresou na jednu společnou sběrnici. V praxi jsou tato čísla však podstatně nižší, protože adresa čipu většinou nelze určit plnými 7 (10) bity ale třeba jen třemi. Někdy nelze určit vůbec a je dána na pevně pro daný typ čipu - takových čipů tedy na jedné sběrnici nemůže být více než jeden. Přenosová rychlost sběrnice je pro většinu aplikací dostatečná i v základní verzi, kde je frekvence hodin 100kHz. Ve vylepšených verzích to může být 400kHz nebo 1MHz, ale ne všechny integrované obvody tuto verzi podporují. Rychlost přenosu pak musí být přizpůsobena pochopitelně "nejpomalejšímu" čipu na sběrnici. Oba vodiče musí být implicitně v logické jedničce a to je zajištěno pull-up rezistory. Jejich odpory mají hodnotu v řádech jednotek kiloohmů. Čím je vyšší komunikační frekvence, tím musí být nižší hodnoty těchto odporů. Pro 100kHz postačuje 4k7.

Pro řízení tuneru budeme používat 7 bitové adresování a maximální přenosovou rychlost 100 kHz. Pro začátek budeme komunikovat pouze jedním směrem a to od RASPI (master) k jednotlivým obvodům připojeným na sběrnici (slave).

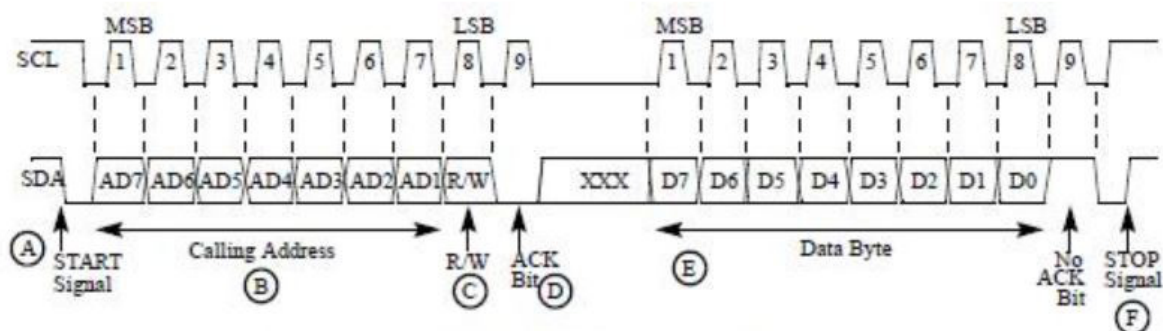
Když si popíšeme jakým způsobem bude master komunikovat, tak to bude vypadat podle následujícího obrázku:



Vysílání začíná vysláním START bitu (A) a následuje 7 bitová adresa obvodu (B). Osmý bit nám určuje, zda budeme data posílat nebo je naopak číst – R/W (C). Logická 0 znamená, že data posílá master, jednička znamená, že master očekává data od obvodu. Po tomto bitu následuje 1 bit ACK (D) - ten vysílá oslovený slave a potvrzuje tím, že je připraven přijímat data. Po té následuje 1. byte dat (E), která opět slave potvrdí signálem ACK a následuje 2. byte dat s potvrzením ACK od slave. Ukončení přenosu master zakončí STOP bitem (F). Datových byte může být více nebo naopak jen jeden, ale posloupnost zůstává zachována.

START	Slave address	Rd/nWr	ACK	Data	ACK	Data	ACK	STOP
1 bit	7 bits	1 bit	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit

• I2C protocol



V případě čtení (R/W=1) data posílá slave a master potvrzuje přijetí signálem ACK.

Při praktické realizaci propojíme modul tuneru tří vodičovým spojením : SDA, SCL a GND. Napájení jednotlivých zařízení je samostatné. Dáme pozor, abychom pull-up odpory na modulu osadili správně, to je podle napájecího napětí mastra. Raspi má napájení 3.3V, tak ho dodržíme a nebudeme raději zkoušet, zda brána I2C vydrží 5V.

Pro úspěšné programování budeme potřebovat základní údaje od jednotlivých obvodů. To lze dohledat v datasheetech obvodů, ale pro lepší přehlednost si je ukážeme v následujících tabulkách.

Adresy obvodů:

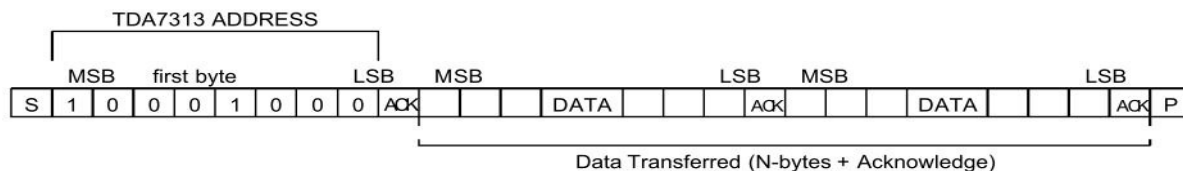
Obvod	b7	b6	b5	b4	b3	b2	b1	b0	HEX	RASPI	Poznámka
TDA7313	1	0	0	0	1	0	0	x	0x88	0x44	
TEA6825	1	1	0	0	0	0	1	x	0xC2	0x61	
TEA6810	1	1	0	0	0	1	0	0	0xC4	0x62	

<http://www.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-practicke-vyuziti-k-pripojeni-externi-eeeprom-24lc256>

TDA7313 – signálový procesor:

Obvod TDA7313 umožňuje řídit hlasitost, nastavovat úroveň výstupu pro reproduktory, přepínat vstupy a řídit hloubky a výšky. Například pro nastavení maximální hlasitosti odešleme:

1. adresu obvodu - 10001000 (0x88)
2. data 1 byte - 00000000 (0x00)



ACK = Acknowledge

S = Start

P = Stop

MAX CLOCK SPEED 100kbits/s

SOFTWARE SPECIFICATION

Chip address

1	0	0	0	1	0	0	0
MSB				LSB			

DATA BYTES

MSB								LSB			FUNCTION
0	0		B2	B1	B0	A2	A1	A0			Volume control
1	1	0		B1	B0	A2	A1	A0			Speaker ATT LR
1	1	1		B1	B0	A2	A1	A0			Speaker ATT RR
1	0	0		B1	B0	A2	A1	A0			Speaker ATT LF
1	0	1		B1	B0	A2	A1	A0			Speaker ATT RF
0	1	0		G1	G0	S2	S1	S0			Audio switch
0	1	1		0	C3	C2	C1	C0			Bass control
0	1	1		1	C3	C2	C1	C0			Treble control

Ax = 1.25dB steps; Bx = 10dB steps; Cx = 2dB steps; Gx = 3.75dB steps

Speaker Attenuators

MSB						LSB			FUNCTION
1	0	0		B1	B0	A2	A1	A0	Speaker LF
1	0	1		B1	B0	A2	A1	A0	Speaker RF
1	1	0		B1	B0	A2	A1	A0	Speaker LR
1	1	1		B1	B0	A2	A1	A0	Speaker RR
						0	0	0	0
						0	0	1	-1.25
						0	1	0	-2.5
						0	1	1	-3.75
						1	0	0	-5
						1	0	1	-6.25
						1	1	0	-7.5
						1	1	1	-8.75
						0	0		0
						0	1		-10
						1	0		-20
						1	1		-30
						1	1	1	Mute

For example attenuation of 25dB on speaker RF is given by:

1 0 1 1 0 1 0 0

Bass and Treble

0	1	1	0	C3	C2	C1	C0	Bass
0	1	1	1	C3	C2	C1	C0	Treble
				0	0	0	0	-14
				0	0	0	1	-12
				0	0	1	0	-10
				0	0	1	1	-8
				0	1	0	0	-6
				0	1	0	1	-4
				0	1	1	0	-2
				0	1	1	1	0
				1	1	1	1	0
				1	1	1	0	2
				1	1	0	1	4
				1	1	0	0	6
				1	0	1	1	8
				1	0	1	0	10
				1	0	0	1	12
				1	0	0	0	14

C3 = Sign

For example Bass at -10dB is obtained by the following 8 bit string:

0 1 1 0 0 0 1 0

Audio Switch

MSB			LSB					FUNCTION
0	1	0	G1	G0	S2	S1	S0	Audio Switch
						0	0	Stereo 1
						0	1	Stereo 2
						1	0	Stereo 3
						1	1	Stereo 4 (*)
					0			LOUDNESS ON
					1			LOUDNESS OFF
			0	0				+11.25dB
			0	1				+7.5dB
			1	0				+3.75dB
			1	1				0dB

For example to select the stereo 2 input with a gain of +7.5dB LOUDNESS ON the 8bit string is:

0 1 0 0 1 0 0 1

(*) Stereo 4 is connected internally, but not available on pins.

Volume

MSB				LSB			FUNCTION	
0	0	B2	B1	B0	A2	A1	A0	Volume 1.25dB steps
					0	0	0	0
					0	0	1	-1.25
					0	1	0	-2.5
					0	1	1	-3.75
					1	0	0	-5
					1	0	1	-6.25
					1	1	0	-7.5
					1	1	1	-8.75
0	0	B2	B1	B0	A2	A1	A0	Volume 10dB steps
		0	0	0				0
		0	0	1				-10
		0	1	0				-20
		0	1	1				-30
		1	0	0				-40
		1	0	1				-50
		1	1	0				-60
		1	1	1				-70

For example a volume of -45dB is given by:

0 0 1 0 0 1 0 0

TEA6825T

In Car Entertainment (ICE) car radio

Zde se nastavují základní parametry tuneru. V prvním byte bit 1. se nastavuje STEREO – MONO, bity 1 až 3 určují krok ladění (50 kHz), bitem 4 nastavujeme zatlumení při ladění, 5.bit nastavuje SDR, 6.bit umlčí celé rádio a 7. bit povoluje posílat povely na TEA6810. Sběrnici pro vstupní obvod je vhodné vypínat po naladění, protože signály na sběrnici I2C by mohly rušit vstupní obvody tuneru. Proto se sběrnice povoluje pouze v případě ladění (změny frekvence).

Červeně zvýrazněné hodnoty nastavíme při prvním programování, později je můžeme nastavovat dle požadavků.

Data to be received by the IC for data byte 1

BIT	DESCRIPTION	RESULT
0	switch for mono	bit 0 = 1
	switch for stereo	bit 0 = 0
1	LSB reference frequency for synthesizer	bit 1 = 1
2	reference frequency for synthesizer	bit 2 = 0
3	MSB reference frequency for synthesizer	bit 3 = 1
4	tuning mute off	bit 4 = 1
	tuning mute on	bit 4 = 0
5	SDS/SDR hold off	bit 5 = 1
	SDS/SDR hold on	bit 5 = 0
6	radio mute off	bit 6 = 1
	radio mute on	bit 6 = 0
7	I ² C-bus to front-end ENABLED	bit 7 = 1
	I ² C-bus to front-end DISABLED	bit 7 = 0

Reference frequency setting in data byte 1;

BIT 3	BIT 2	BIT 1	FREQUENCY SETTING
0	0	0	3 kHz
0	0	1	5 kHz
0	1	0	10 kHz
0	1	1	15 kHz
1	0	0	25 kHz
1	0	1	50 kHz
1	1	0	not defined
1	1	1	not defined

Data to be received by the IC for data byte 2

BIT	DESCRIPTION	RESULT
0	AM/FM; AM mode	bit 0 = 0
	AM/FM; FM mode	bit 0 = 1
1	divider for AM mixer; divide by 2	bit 1 = 0
	divider for AM mixer; divide by 6	bit 1 = 1
2	measure time IF-count; 40 ms	bit 2 = 0
	measure time IF-count; 4 ms	bit 2 = 1
3	SDR off	bit 3 = 0
	SDR on	bit 3 = 1
4	IF-prescaler division rate; divide by 200	bit 4 = 0
	IF-prescaler division rate; divide by 25	bit 4 = 1
5	sensitivity unchanged	bit 5 = 0
	multi-path sensitivity switch; less sensitivity by an offset of $\Delta m = 10\%$	bit 5 = 1
6	temperature compensation of unweighted voltage level; temperature coefficient as specified in Chapter "Characteristics"	
	temperature compensation off	bit 6 = 0
7	temperature compensation on	bit 6 = 1
	noise blanker deactivated by high modulation	bit 7 = 0
	noise blanker not deactivated	bit 7 = 1

TEA6810V

Front-end and PLL synthesizers for car radios

Příkazy pro vstupní obvod TEA6810 se posílají v následující sekvenci:

1. Adresa obvodu
2. Nižší a vyšší byte dělicího poměru pro ladění - 0x0E 0x0D (95.0 MHz) *
3. Byte pro spínání (FM/AM, dělička AM, DX, a další)
4. Testovací byte (0x00)

Table 1 Bit organization

START	MODULE ADDRESS	PROGRAMMABLE DIVIDER DATA					SWITCH CONTROL	TEST			STOP
S	byte 0	A	byte 1	A	byte 2	A	byte 3	A	byte 4	A	P

Table 2 I²C-bus address and received bytes

BYTES TO BE RECEIVED (4 BYTES)	BUS ADDRESS							
	MSB				LSB			
	1	1	0	0	0	1	0	0
Byte 1 ⁽¹⁾ program divider N1 (Low byte)	S7	S6	S5	S4	S3	S2	S1	S0
Byte 2 ⁽¹⁾ program divider N1 (High byte)	S15	S14	S13	S12	S11	S10	S9	S8
Byte 3 switching	MSB in-lock counter 0	LSB in-lock counter 0	0 = LOW current tuning oscillator 1 = HIGH	0 = LOW current charge pump 1 = HIGH	1 = FM local 0 = FM dx	MSB divider N2 1	LSB divider N2 1	0 = AM 1 = FM
Byte 4 testing	1 = 3-state 0 = normal charge pump	1 = f _{div} 0 = LCKDET	1 = test 0 = normal in-lock counter	X ⁽²⁾ 0	X 0	X 0	X 0	X 0

Table 3 N2 divider

N2 DIVIDER	MSB	LSB
3	0	0
5	0	1
10	1	0
15	1	1

Table 4 In-lock

IN-LOCK	MSB	LSB	AM/FM
8	0	0	FM
16	1	0	AM or FM
32	1	1	AM or FM
48	0	0	AM
64	0	1	AM

* Požadovaná frekvence se nastavuje v byte 1 a 2 následovně:

Frekvenci v kHz vynásobíme 2 a připočteme konstantu 1442, což je hodnota mezifrekvence při ladění po 50 kHz.

Příklad: chceme nastavit rádio Blaník na 95.0 MHz ($950 \cdot 2 + 1442 = 3342 \Rightarrow$ převedeme na hexadecimální tvar $\Rightarrow 0x0D0E$. První byte tedy bude 0x0E, druhý byte 0x0D.

Příklad pro ovládání I2C na Raspberry PI, který je převzatý:

<http://www.raspi.cz/2012/11/i2c-potreتي-rozsvecime-led-diodu-sloziteji-draze/>

a slouží k ovládání LED.

Po rebootu se RPi zeptáme, zda je tam něco nového na I2C sběrnici.

```
pi@raspberrypi ~ $ i2cdetect -y 0
```

```
0 1 2 3 4 5 6 7 8 9 a b c d e f
```

```
00: _____
```

```
10: _____
```

```
20: 20 _____
```

```
30: _____
```

```
40: _____ 48 _____ 4d _____
```

```
50: _____
```

```
60: _____
```

```
70: _____
```

Na adrese **0x20** vidíme nové zařízení. 0x48 a 0x4d jsou teplotní čidla z minulého dílu.

Nastavíme registr IODIRA (0), který určuje, zda jsou piny vstupní nebo výstupní. 1 = vstup, 0 = výstup. Pro komunikaci použijeme **i2cset**, kterému předáme parametry:

- -y = neptej se na nic
- 1 = sběrnice 1
- 0x20 = adresa čipu na sběrnici
- 0 = adresa registru v čipu
- 0xfe = hodnota, co nastavujeme (1111 1110)
- b = zapisujeme 1 byte

```
pi@raspberrypi ~ $ i2cset -y 1 0x20 0 0xfe b
```

LED se rozvítla! Defaultní hodnota výstupního pinu je totiž „0“.

No a teď můžeme blikat LEDkou. Piny se ovládají v registru OLATA (0x0a).

```
pi@raspberrypi ~ $ i2cset -y 0 0x20 0x0a 0x01 b
```

(1 = zhasnuto)

```
pi@raspberrypi ~ $ i2cset -y 0 0x20 0x0a 0x00 b
```

(0 = svítí)

```
pi@raspberrypi ~ $ i2cset -y 0 0x20 0x0a 0x01 b
```

Ovládání tuneru se provádí obdobně, to znamená k příkazu s parametry **\$ i2cset -y 0** přidáme adresu obvodu a následuje 1 byte dat.

Mnoho zdaru v experimentování.

Zkušenosti s Raspberry 2B

**) Oproti minulým verzím má RP instalovanou sběrnici I2C defaultně 1 !*

Druhá důležitá poznámka je, že RPI2 má osazené pullup odpory již na desce, takže neosazujeme

na naší desce žádný z odporů R1,R2,R112 a R113 !

A třetí: u TDA7313 je třeba připojit pin 3 na GND (propojka vypadla v návrhu) !

Příkaz **i2cdetect -y 1** vypíše známá zařízení na sběrnici.

Vypisuje čísla aktivních čipů na sběrnici. Příkaz to dělá tak, že osloví všechny adresy postupně a která mu odpoví signálem ACK, tak ji bere jako aktivní zařízení. Ale stačí také to, že z nějakého důvodu máme signál SDA na nule a příkaz vypíše, že aktivní čipy jsou na všech adresách. Když je sběrnice v pořádku vypisuje jen aktivní čipy.

Jako první při ožiování bychom měli mít adresu 0x61. Nenechte se zmýlit, že v datasheetu má TEA6825 adresu 0xC2 (B'11000010'). Je to stejné, jen RP má adresování do 0x77 (7 bit adresování) a dochází tady k posunu hodnot jednotlivých bitů doprava. Vypadnul nám nultý bit a ostatní se posunuly jako bychom provedli instrukci rotace right, což je vlastně dělení 2. Takže když se vrátíme k původní adrese z datasheetu 0xC2 a podělíme ji 2, dostaneme skutečně adresu 0x61.

Obdobně je to i u ostatních obvodů:

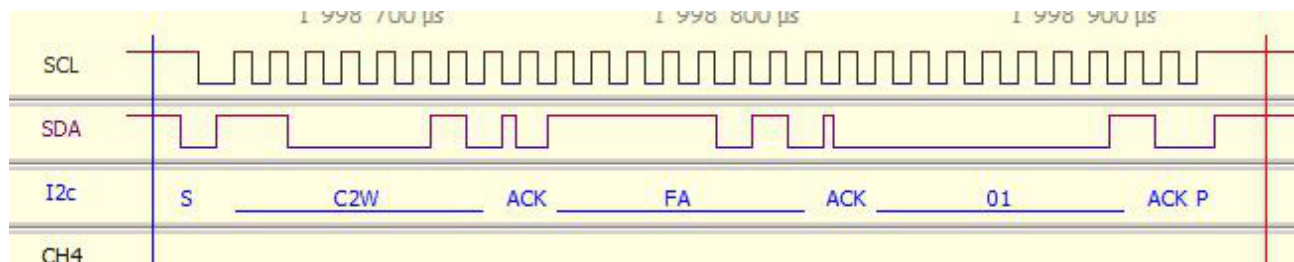
TEA6810 (0xC4) by se měl hlásit jako 0x62 (pozor, I2C pro něho musíme nejprve povolit příkazem na TEA6825)

TDA7313 (0x88) by se měl hlásit jako 0x44 a těmito adresami je také budeme ovládat.

Takže zpět k příkazu pro TEA6825:

vypíšeme příkaz **i2cset -y 1 0x61 0xFA 0x01 i** a odešleme ENTER.

Následující obrázek znázorňuje, co bylo odesláno na sběrnici I2C:



Signál SCL jsou hodiny a SDA jsou data. Relace začíná S – Start bit, následuje adresa čipu a že se jedná o zápis (C2W), následuje ACK (odpověď čipu) FA je první odeslaný byte a 01 je druhý odeslaný byte. Po každém byte je potvrzovací signál ACK od čipu a relace je ukončena (P).

Ještě k parametrům příkazu:

- **i2cset** příkaz
- **-y** parametr „na nic se neptej“
- **1** číslo sběrnice I2C v Raspi
- **0x61** adresa čipu TEA6825
- **0xFA** 1.byte
- **0x01** 2.byte
- **i** parametr I2C block data

Po tomto příkazu by TEA6825 měl povolit I2C pro TEA6810, takže po příkazu: **i2cdetect -y 1** by měl počítač vypsat následující tabulku aktivních zařízení připojených na I2C:

pi@raspberrypi ~ \$ i2cdetect -y 1

```
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- 44 -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- 61 62 -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

Vše je v pořádku, můžeme pokračovat. Lze ovládat celkem vše, tak jen pro ilustraci některé funkce.

TDA7313 (0x44)

Inicializace:

spočívá v nastavení atenuátoru reproduktorů v pořadí LF,RF,LR a RR.

i2cset -y 1 0x44 0x80 0xA0 0xC0 0x E0 i je příkaz složený ze 4 byte, kde každý byte nastavuje jeden kanál. Povolený rozsah je 1F, kde 0 je maximální úroveň a 1F je maximální zatlumení (38,75 dB) s krokem 1,25 dB. Parametr **i** znamená I2C blok data, **c** je byte data

Nastavení hlasitosti :

i2cset -y 1 0x44 0x1A c je příkaz pro nastavení hlasitosti TDA7313 na úroveň -17,5 dB

0x1A je byte pro ovládání hlasitosti a může být v rozsahu 0x00 až 0x3F, kde 0 je maximální hlasitost a 3F maximální zatlumení (-78,75 dB) Krok je 1,25 dB.

Přepnutí vstupů :

i2cset -y 1 0x44 0x58 c je příkaz pro přepnutí vstupu 1 s nulovým zesílením.

Lze nastavit maximálně vstup 1, 2 a 3 vždy nejnižšími dvěma bity. Čtvrtý vstup lze nastavit, ale jen vnitřně, protože není vyveden na piny, takže jako by nebyl.

TEA6825 (0x61)

Inicializace:

spočívá v nastavení požadovaných parametrů. Posílají se 2 byte.

i2cset -y 1 0x61 0xFA 0x01 i

Nejvyšším bitem prvního byte se povoluje I2C pro přední IO tuneru (TEA6810). Je to z toho důvodu, aby povely procházející po sběrnici nerušily analogovou část vstupu tuneru (rušení). Nejnižší bit je přepínač mono – stereo. Ostatní bity jsou určeny pro nastavení frekvence syntetizéru, povely mute pro rádio a pro ladění. Pro začátek není dobré hodnotu 0xFA měnit. Druhý byte je dobré nechat nulový, kromě nejnižšího bitu, kde se nastavuje mód FM.

TEA6810 (0x62)

Inicializace:

se provádí celkem ve 4 byte:

1.byte je „Low byte“, další je „Hight byte požadované frekvence

3.byte jsou nastavení obdobná předcházejícímu čipu. Důležitý je nejnižší bit – přepínač AM-FM.

4.byte je testovací – nastavuje se na nulu.

`i2cset -y 1 0x62 0x0E 0x0D 0x37 0x00 i` - 0x0D0E je frekvence 95.00 MHz.

Ladění:

`i2cset -y 1 0x62 0x0E 0x0D i` - povolený rozsah pro frekvenci FM rádia (87,5 – 108 MHz) je v rozmezí 3192 až 3602 (0x0C78 – 0x0E12) pro ladění po 50kHz, které máme nastaveno. Takže pro náš příklad rádia Blaník na 95.0 MHz je $(950 \cdot 2 + 1442 = 3342 \Rightarrow 0x0D0E)$.

Při ladění se dají posílat pouze dva byte požadované frekvence, nesmíme však zapomínat, že si musíme povolit sběrnici I2C na TEA6825 (adresa 0x61) a že posíláme nejdříve nižší a po té vyšší byte.

Sekvence příkazů pro aktivaci tuneru FM – příklad:

`i2cset -y 1 0x44 0x80 0xA0 0xC0 0x E0 i` - inicializace TDA7313

`i2cset -y 1 0x61 0xFA 0x01 i` - inicializace TEA6825

`i2cset -y 1 0x62 0x0E 0x0D 0x37 0x00 i` - inicializace TEA6810 (95,0MHz)

`i2cset -y 1 0x44 0x58 c` - přepnutí nf vstupu (Stereo1)

`i2cset -y 1 0x44 0x1A c` - nastavení hlasitosti

`i2cset -y 1 0x62 0x9C 0x0C i` - ladění frekvence (89,3MHz)