

## FM Radio Receiver Module (#27984)

The FM Radio Receiver Module uses an RDA5807SS FM stereo radio tuner chip, which provides an easy way for your microcontroller to receive local FM radio stations. Using the onboard antenna and headphone jack, you can easily create your own portable radio!

### Features

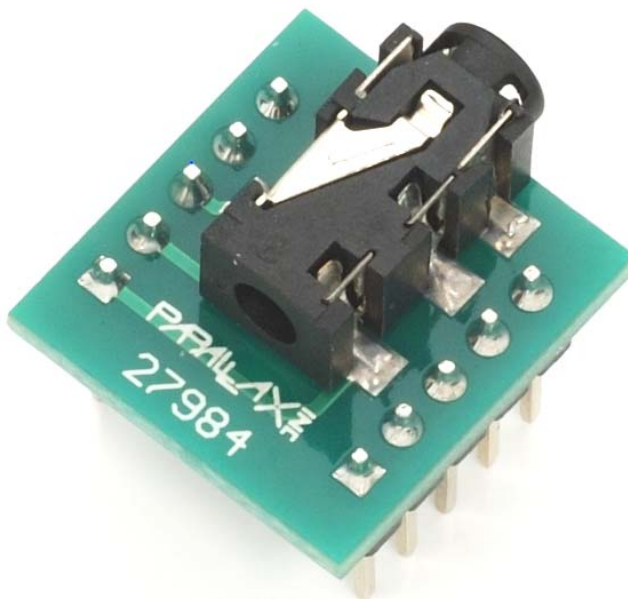
- Onboard standard 3.5 mm TRS stereo headphone minijack provides easy audio output
- Supports the worldwide frequency band (76 to 108 MHz)
- No external antenna required
- 10-pin DIP package ready for breadboard or through-hole projects
- Easy I<sup>2</sup>C interface with virtually any microcontroller

### Key Specifications

- Power Requirements: 2.7 to 5.5 VDC  
@ 25 mA
- Communication: I<sup>2</sup>C
- Operating temperature: 32 to 121 °F  
(0 to 50 °C)
- Dimensions: 0.70 x 0.65 x 0.34 in  
(1.78 x 1.65 x 0.87 cm)

### Application Ideas

- Portable radio
- Digital radio alarm clock

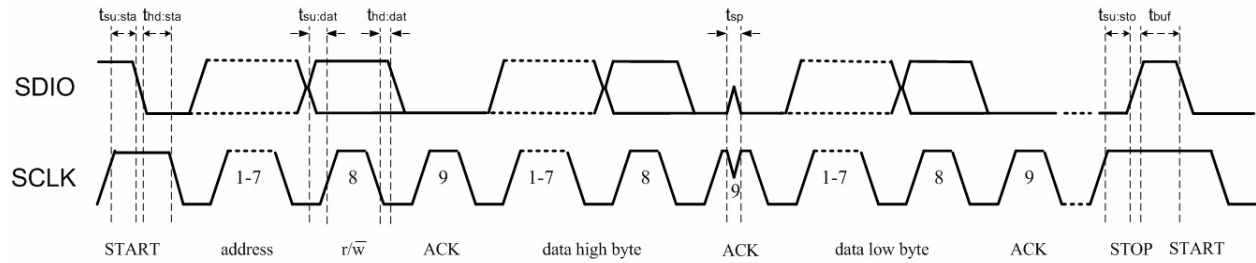


### Theory of Operation

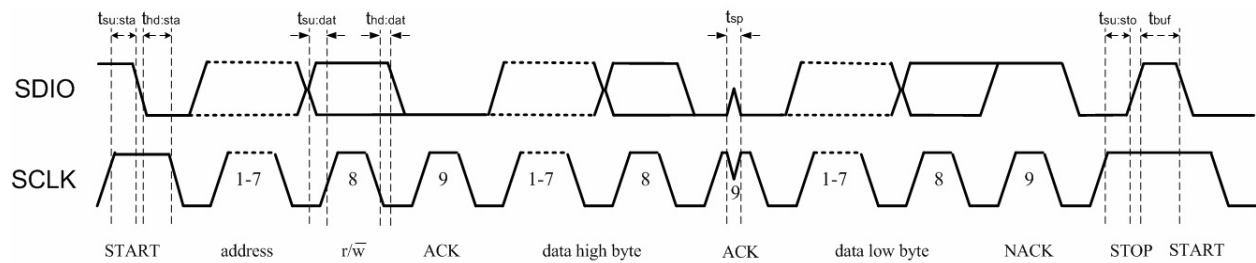
The FM Radio Receiver Module uses the RDA5807SS FM stereo radio tuner chip to tune to FM radio stations on the worldwide frequency band. You can program this chip directly using an I<sup>2</sup>C interface through the SDA and SCL pins (for more detailed information, see the Register Table section on page 3).

An I<sup>2</sup>C interface transfer begins with a START condition, a command byte and data bytes. Each byte is followed by an ACK (or NACK) bit, and then the interface transfer ends with a STOP condition. The command byte includes a 7-bit chip address and a Read/Write bit. The ACK (or NACK) is always sent out by the receiver. During a write transaction, data bytes are sent from the microcontroller, and during a read transaction, data bytes are read from the FM Radio Receiver Module.

The FM Radio Receiver Module will send an ACK after every byte, and the microcontroller should issue a STOP condition when register programming is finished. Detailed timing diagrams can be seen in Figure 1 and Figure 2 below.



**Figure 1: Write Transaction Timing Diagram**



**Figure 2: Read Timing Diagram**

The FM Radio Receiver Module also features an onboard stereo headphone jack. When headphones are plugged in, the ground wire of the headphone's cable functions as an antenna. However, should better reception be needed, and external antenna can be connected to the FM Radio Receiver Module's ANT pin.

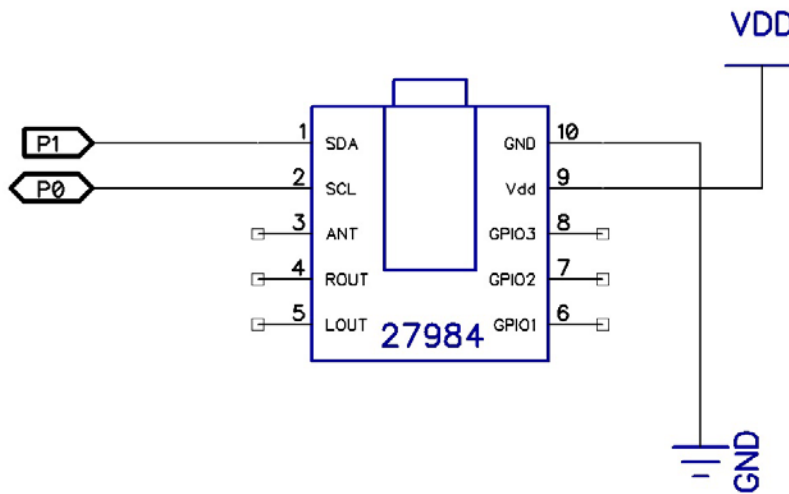
## Pin Definitions and Ratings

Pin	Name	Type	Function
1	SDA	I/O	Serial Data Line
2	SCL	I	Serial Clock Input (0 to 400 kHz)
3	ANT	-	External Antenna Connection Option
4	ROUT	O	Right Audio Output
5	LOUT	O	Left Audio Output
6	GPIO1	I/O	General Purpose Input/Output 1
7	GPIO2	I/O	General Purpose Input/Output 2
8	GPIO3	I/O	General Purpose Input/Output 3
9	VDD	P	Supply Voltage -> 2.7 to 5.5 VDC
10	GND	G	Ground -> 0 V

Pin Type: P = Power, G = Ground, I = Input, O = Output

## Connection Diagram

For use with the example programs included on the FM Radio Receiver Module product page. Go to [www.parallax.com](http://www.parallax.com) and search 27984.



## Register Table

Communicating with the FM Radio Receiver Module is done through an I2C interface. A write operation is defined as data going from a microcontroller to the FM Radio Receiver Module, and a read operation is defined as data going from the FM Radio Receiver Module to a microcontroller. Below is a table of the most commonly used registers, their functions and default values.

Register	Bit	Name	Function	Default
0x02	15	DHIZ	Audio Output 0 = High Impedance; 1 = Normal Operation	0
	14	DMUTE	Mute Disable 0 = Mute; 1 = Normal Operation	0
	13	MONO	Mono Select 0 = Stereo; 1 = Force Mono	0
	12	BASS	Bass Boost 0 = Disabled; 1 = Bass boost enabled	0
	11	RESERVED	-	-
	10	CLK32_INPUT_ENB	0 = Enable; 1 = Disable (select 1 if crystal is shared)	0
	8:7	RESERVED	-	-
	6:4	CLK_MODE[2:0]	000 = 32.768 kHz (used for this module)	000
	1	SOFT_RESET	Soft reset; set to 1 to reset.	0
	0	ENABLE	Power Up Enable 0 = Disabled; 1 = Enabled	0

Register	Bit	Name	Function	Default
0x03	15:6	CHAN[9:0]	Channel Select BAND = 0 Frequency = Channel Spacing (kHz) x CHAN + 87 MHz BAND = 1 Frequency = Channel Spacing (kHz) x CHAN + 76 MHz <i>Updated after a seek operation.</i>	0x00
	5	RESERVED	-	-
	4	TUNE	Tune 0 = Disable; 1 = Enable	0
	3:2	BAND[1:0]	Band Select 00 = 87 – 108 MHz (US/Europe) 01 = 76 – 91 MHz (Japan) 10 = 76 – 108 MHz (Japan wide)	00
	1:0	SPACE[1:0]	Channel Spacing 00 = 100 kHz 01 = 200 KHz 10 = 50 kHz 11 = 12.5 kHz	00
0x04	14	STCEIN	Seek/Tune Complete Interrupt Enable 0 = 1 Disable Interrupt; 1 = Enable Interrupt <i>Setting to 1 will generate a low pulse on GPIO2 when the interrupt occurs.</i>	0
	12	RESERVED	-	-
	11	DE	De-emphasis 0 = 75 µs; 1 = 50 µs	0
	9:7	RESERVED	-	-
	6	I2S_ENABLED	I2S bus enable 0 = disabled; 1 = enabled	0
	5:4	GPIO3[1:0]	General Purpose I/O 3 00 = High Impedance 01 = Mono/Stereo indicator (ST) 10 = Low 11 = High	00
	3:2	GPIO2[1:0]	General Purpose I/O 2 00 = High Impedance 01 = Interrupt (INT) 10 = Low 11 = High	00
	1:0	GPIO1[1:0]	General Purpose I/O 1 00 = High Impedance 01 = Reserved 10 = Low 11 = High	00

Register	Bit	Name	Function	Default
0x05	15	INT_MODE	If 0, generate 5ms interrupt; If 1, interrupt lasts until read register 0x0C occurs	1
	14:8	SEEKTH[6:0]	Seek Threshold RSSI scale is logarithmic, 0000000 = min RSSI	0001000
	7:6	LNA_PORT_SEL[1:0]	LNA input port selection bit 00 = no input 01 = LNAN 10 = LNAP 11 = dual port input	10
	5:4	LNA_ICSEL_BIT[1:0]	LNA working current bit: 00 = 1.8 mA 01 = 2.1 mA 10 = 2.5 mA 11 = 3.0 mA	10
	3:0	VOLUME[3:0]	DAC Gain Control Bits (Volume) 0000 = min; 1111 = max (Scale is logarithmic)	1000
0x0A	15	RESERVED	-	-
	14	STC	Seek/Tune Complete 0 = Not complete; 1 = Complete <i>Flag is set when the seek or tune operation completes.</i>	0
	13	SF	Seek Fail 0 = Seek successful; 1 = Seek failure <i>Flag is set when the seek operation fails to find a channel with an RSSI level greater than SEEKTH[5:0]</i>	0
	12:11	RESERVED	-	-
	10	ST	Stereo Indicator 0 = Mono; 1 = Stereo <i>Stereo indication available on GPIO3.</i>	1
	9:0	READCHAN[9:0]	Read Channel BAND = 0 Frequency = Channel Spacing (kHz) x READCHAN[7:0] + 87 MHz BAND = 1 Frequency = Channel Spacing (kHz) x READCHAN[7:0] + 76.0 MHz <i>READCHAN[7:0] is updated after a tune or seek operation.</i>	0x00000000
0x0B	15:9	RSSI[6:0]	RSSI (scale is logarithmic) 000000 = min; 111111 = max	000000
	8	FM_TRUE	0 = Current channel is not a station 1 = Current channel is a station	0
	7	FM_READY	0 = not ready; 1 = read <i>Used for soft seek</i>	0

## BASIC Stamp® Example Code

Included on the FM Radio Receiver Module product page is demo code which transforms your BASIC Stamp into a FM Stereo! Listen to your favorite stations from your desk using the user interface in the Debug Terminal. You can download this code from the FM Radio Receiver Module product page (ParallaxFMRadio.bs2).

```
66
|

Parallax FM Radio
-----
Bass                                RSSI : 42
Stereo                               Volumn: 12
FM 92.5 MHz
-----

Select:

7)Vol +      8)Freq +      9)Channel+
4)Search -   5)Bss&Str     6)Search +
1)Vol -      2)Freq -      3)Channel-
0)Power      .)Mute      Enter) Save

Save FM Stations:

12) 94.4 MHz
11) 95.9 MHz
10) 94.7 MHz
09) 95.2 MHz
08) 93.5 MHz
07) 95.2 MHz
06) 94.7 MHz
05) 93.6 MHz
04) 94.0 MHz
03) 101.1 MHz
02) 96.9 MHz
01) 92.5 MHz    <- Save at

Macros...  Pause  Clear  Close   Echo Off
```

## Propeller™ P8X32A Example Code

Included on the FM Radio Receiver Module product page is demo code which transforms your Propeller into a FM Stereo! Listen to your favorite stations from your desk using the user interface in the Parallax Serial Terminal. You can download this code from the FM Radio Receiver Module product page (ParallaxFMRadio\_Spin.zip). (This demo uses the Parallax Serial Terminal software which is included with the Propeller Tool software installer.)

