RDA1846 Programming Guide

Contents

Document overview

This programming guide has been restructured from previous revisions for clarity. This contains two documents for interface and programmer separately. Interface document contains I2C interface, 3 wire SPI interface and 4 wire SPI interface .Programmer document contains a complete programming guide for using any interface.

Doc. A: Interface

RDA1846 each register write is 24-bit long, including a r/$\overline{W}$ bit,7-bit register address , and 16-bit data (MSB is the first bit).

R/W	A[6:0]	D[15:0]

Note

If register address is more than 7FH, first write 0x0001 to 7FH, and then write value to the address subtracted by 80H. Finally write 0x0000 to 7FH

Example: writing 85H register address is 0x001F .
Move 7FH 0x0001;
Move 05H 0x001F; 05H=85H-80H
Move 7FH 0x0000;

1. I2C Interface

RDA1846 enable software programming through I2C interface. Software controls chip working states, such as Txon or Rxon operation, and reads status register to get operation result through I2C interface.

It includes two pins: SCLK and SDIO.

A I2C interface transfer begins with START condition, a command byte and data bytes, each byte has a followed ACK (or NACK) bit, and ends with STOP condition. The command byte includes a 7-bit chip address and a r/$\overline{w}$ bit. The 7-bit chip address is 7'b0101110 when SEN is high, or is 7'1110001 when SEN is low.The ACK (or NACK) is always sent out by receiver. When in write transfer, data bytes is written out from MCU, and when in read transfer, data bytes is read out from RDA1846.

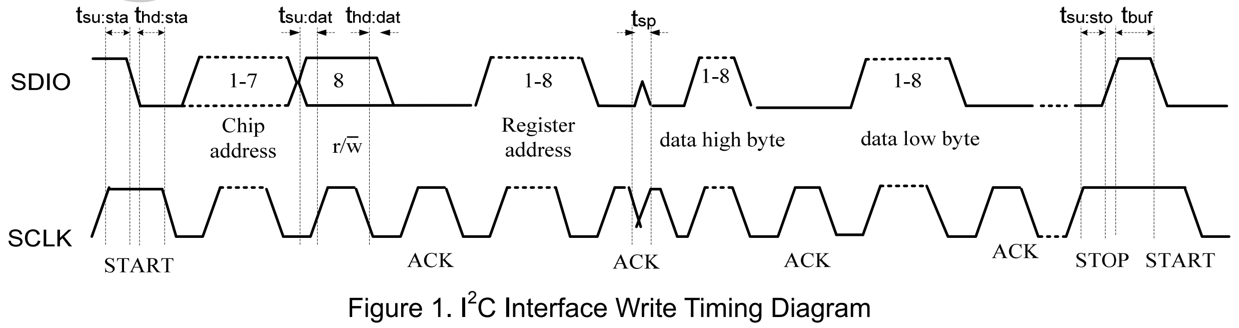


Figure 1. I^{2}C Interface Write Timing Diagram

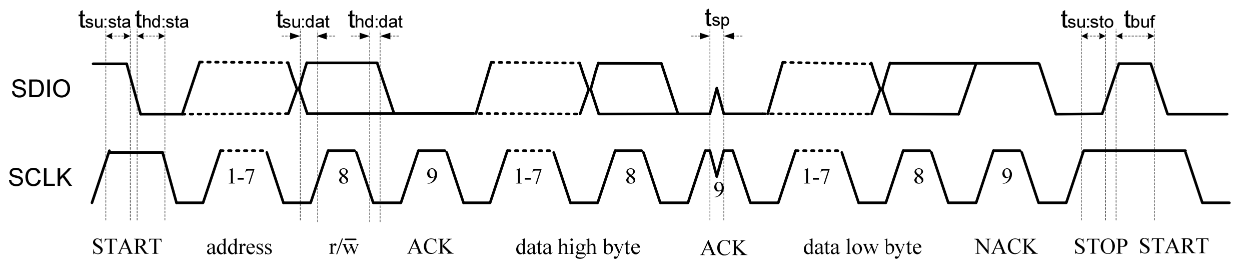


Figure 2. I^{2}C Interface Read Timing Diagram

START	I2C CHIP ADDRESS	W	A	REGISTER ADDRESS	A	REGISTER BIT< 15:8>	A	REGISTER BIT< 7:0>	A/ NA	STOP

Figure 3 I^{2}C Interface Write Combined Format

START	I2C CHIP ADDRESS	W	A	REGISTER ADDRESS	A/ NA	START	I2C CHIP ADDRESS	R	A	REGISTER BIT< 15:8>	A	REGISTER BIT< 7:0>	NA	STOP

Figure 4 I^{2}C Interface Read Combined Format

▨	From master to slave	A = acknowledge (SDA LOW)	S= START condition
□	From slave to master	NA = not acknowledge (SDA HIGH)	P= STOP condition

Table 2. I2C Timing Characteristics

PARAMETER	SYMBOL	TEST CONDITION	MIN	TYP	MAX	UNIT
SCLK Frequency	f_{scl}		0	-	400	KHz
SCLK High Time	t_{high}		0.6	-	-	µs
SCLK Low Time	t_{low}		1.3	-	-	µs
Setup Time for START Condition	$t_{su:sta}$		0.6	-	-	µs
Hold Time for START Condition	$t_{hd:sta}$		0.6	-	-	µs
Setup Time for STOP Condition	$t_{su:sto}$		0.6	-	-	µs
SDIO Input to SCLK↑ Setup	$t_{su:dat}$		100	-	-	ns
SDIO Input to SCLK↓ Hold	$t_{hd:dat}$		0	-	900	ns
STOP to START Time	t_{buf}		1.3	-	-	µs
SDIO Output Fall Time	$t_{f:out}$		$20+0.1C_b$	-	250	ns
SDIO Input, SCLK Rise/Fall Time	$t_{r:in}$ / $t_{f:in}$		$20+0.1C_b$	-	300	ns
Input Spike Suppression	t_{sp}		-	-	50	ns
SCLK, SDIO Capacitive Loading	C_b		-	-	50	pF
Digital Input Pin Capacitance					5	pF

2 Three- wire SPI interface

RDA1846 enable software programming through three-wire(SPI) interface. Software controls chip working states, such as Txon or Rxon operation, and reads status register to get operation result through three-wire interface.

Three-wire interface is slave interface. It includes three pins: $\overline{\text{SEN}}$, SCLK and SDIO. $\overline{\text{SEN}}$ and SCLK are input pins , SDIO are bi-direction pins.

RDA1846 samples command byte and data at posedge of SCLK.The turn around cycle between command byte from MCU and data from RDA1846 is a half cycle. RDA1846 samples command byte at posedge of SCLK, and output data also at posedge of SCLK.

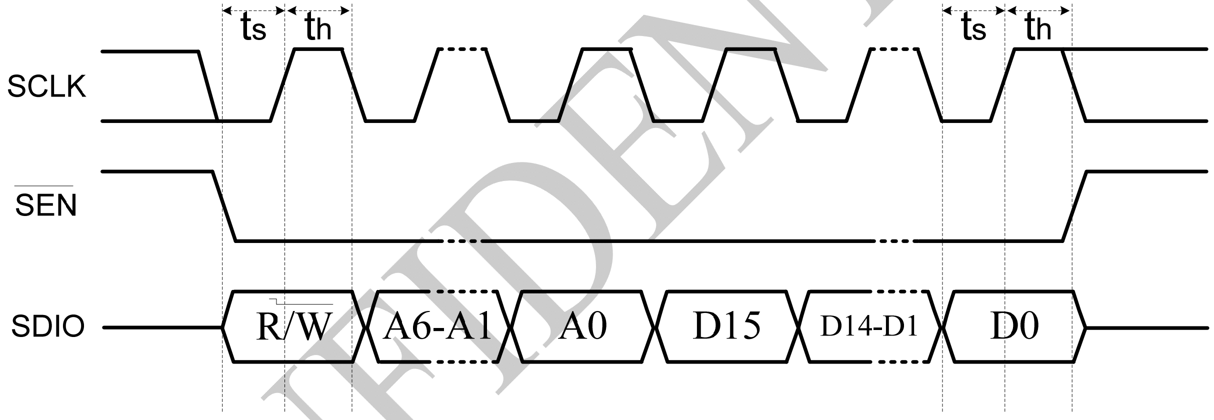


Figure5. Three-wire Interface Write Timing Diagram

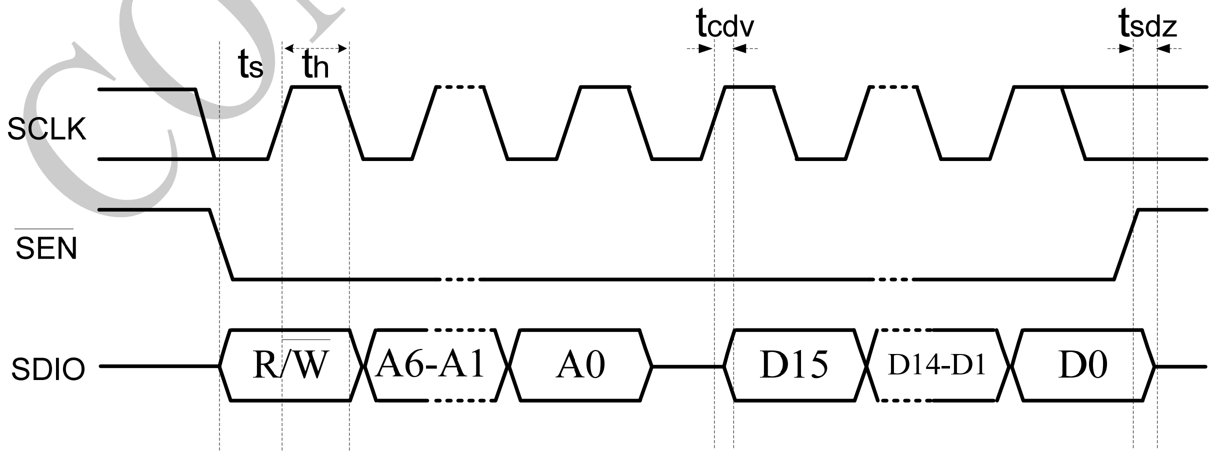


Figure6. Three-wire Interface Read Timing Diagram

Table 2. Three-wire Timing Characteristics

PARAMETER	SYMBOL	TEST CONDITION	MIN	TYP	MAX	UNIT
SCLK Cycle Time	t_{CLK}		35			ns
SCLK Rise Time	t_R				50	ns
SCLK Fall Time	t_F				50	ns
SCLK High Time	t_{HI}		10			ns
SCLK Low Time	t_{LO}		10			ns
SDIO Input, $\overline{\text{SEN}}$ to SCLK↑ Setup	t_s		10	-	-	ns
SDIO Input, to SCLK↑ Hold	t_h		10	-	-	ns
SCLK↑ to SDIO Output Valid	t_{cdv}	Read	2	-	10	ns
$\overline{\text{SEN}}$↑ to SDIO Output High Z	t_{sdz}	Read	2	-	10	ns
Digital Input Pin Capacitance					5	pF

3. Four- wire SPI interface

RDA1846 enable software programming through four-wire(SPI) interface. Software controls chip working states, such as Txon or Rxon operation, and reads status register to get operation result through four-wire interface.

Four-wire interface is slave interface. It includes four pins: $\overline{\text{SEN}}$, SCLK , SDI and SDO. $\overline{\text{SEN}}$,SCLK and SDI are input pins , SDO are bi-direction pins.

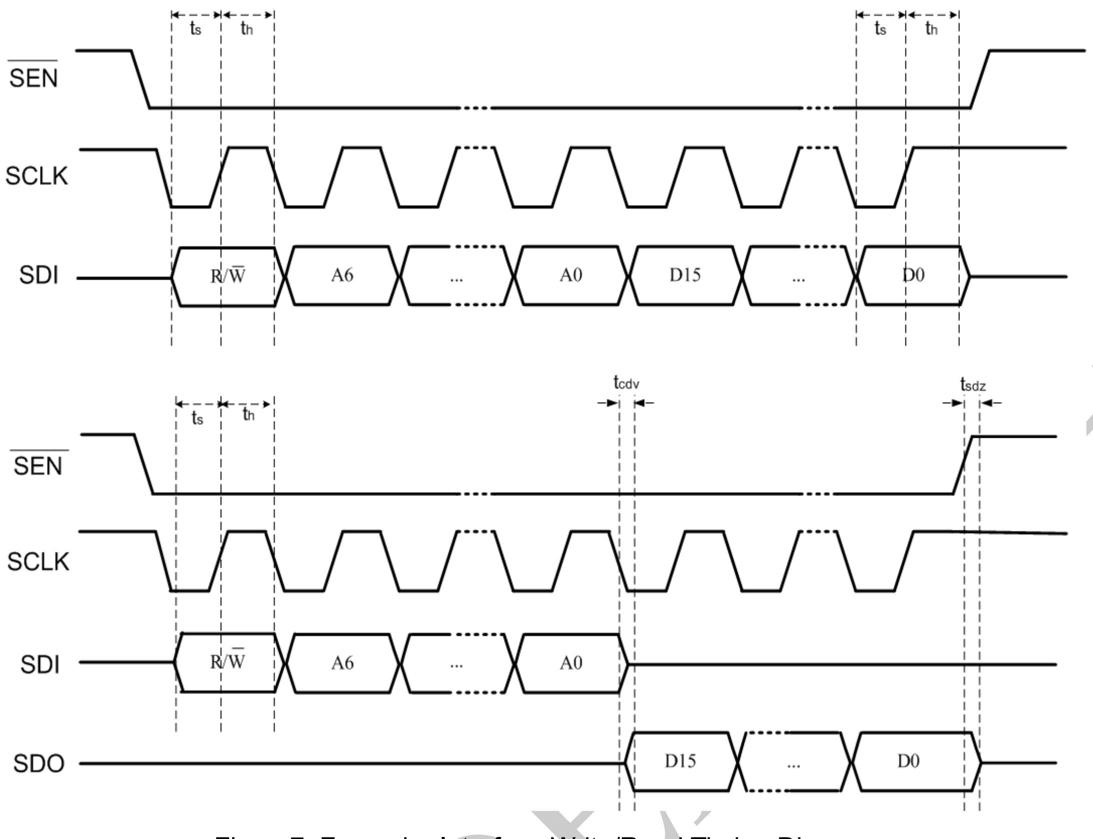


Figure7. Four-wire Interface Write/Read Timing Diagram

Doc. B: Programming guide

1. Setting frequency

Bit	Name	Function
29H[13:0]	freq<29:16>	Freq high value (unit 1khz/8)
2aH[15:0]	freq<15:0>	Freq low value (unit 1khz/8)

Freq<29:0>= Binary (Freq(MHz)*1000*8)
Such as frequency is 409.75MHz, Freq<29:0>=409.75*1000*8=3278000= Binary (1100100000010010110000)
so write 29H [15:0] =0000000000000110010 and 2aH [15:0] = 0000010010110000.

2. Setting RF band

Bit	Name	Function
0fH[7:6]	band_select<1:0>	00 = 400~520MHz 10 =200~260MHz 11 = 134~174MHz

3. Reference clock

RDA1846 takes 12MHz~14MHz or 24MHz~ 28 MHz crystals as its master reference clock. Setting 2bH[15:0], 2cH[15:0] and 04H[0] according different reference clock.

Bit	Name	Function
2bH[15:0]	xtal_freq<15:0>	Crystal clk freq (unit khz) 12~14MHz:crystal freq*1000 24~ 28MHz: (crystal freq/2)*1000
2cH[15:0]	adclk_freq<15:0>	Adc clk freq (unit khz) 12~14MHz:(crystal freq/2)*1000 24~ 28MHz: (crystal freq/4)*1000
04H[0]	clk_mode	12~14MHz:1 24~ 28MHz:0

Such as 12.8M crystal (12MHz~14MHz)

2bH[15:0]= xtal_freq<15:0>=12.8*1000=12800
2cH[12:0] =adclk_freq<15:0>=(12.8/2)*1000=6400
04H[0]= clk_mode =1

26M crystal (24MHz~28MHz)
2bH[15:0]= xtal_freq<15:0>=(26/2)*1000=13000
2cH[15:0] =adclk_freq<15:0>=(26/4)*1000=6500
04H[0]= clk_mode =0

4. Setting Tx and Rx

Bit	Name	Function
30H[13:12]	channel_mode	11 = 25khz channel mode 00 = 12.5khz channel mode 10,01=reserved
30H[6]	tx_on	1 = on 0 = off
30H[5]	rx_on	1 = on 0 = off

5. Deep sleep

Bit	Name	Function
30H[2]	pdn_reg	The same as pdn pin 1 = enable 0 = disable

While Normal mode, pdn_reg and PDN pin must be high at the same time. Only one of pdn_reg and PDN pin is low ,which can turn into deep sleep.

6. TX voice channel

Bit	Name	Function
3cH[15:14]	voice_sel<1:0>	=00; Tx voice signal from MIC =01; Tx inner sine tone setted by tone2 =10; Tx code from GPIO1 code_in (gpio1<1:0> must be set to 01) =11; not Tx any signal

7. TX Pa_bias output voltage

RDA1846 Pa_bias pin output voltage can be controlled by 0aH [5:0].

Bit	Name	Function
0aH [5:0]	pabias_voltage<5:0>	000000: 1.01V 000001:1.05V 000010:1.09V 000100: 1.18V 001000: 1.34V 010000: 1.68V 100000: 2.45V 1111111:3.13V

8. Subaudio

Bit	Name	Function
45H[2:0]	c_mode<2:0>	Ctcss/cdcss mode sel x00=disable, 001=inner ctcss en, 010= inner cdcss en 101= outter ctcss en, 110=outter cdcss en others =disable
45H[3]	ctcss_sel	1 = ctcss_cmp/cdcss_cmp out via gpio 0 = ctcss/cdcss sdo out vio gpio
45H[4]	cdcss_sel	24/23 bit cdcss code sel for both txon and rxon 1 = 24 bit code 0 = 23 bit code
45H[7]	neg_det_en	If 1,cdcss inverse code will be detected at the same time.
45H[11]	Pos_det_en	If 1, cdcss code will be detected.
45H[10]	css_det_en	If 1, sq detection will add ctcss/cdcss detect result, then 1846 control 1846 voice output on or off.
4aH[15:0]	ctcss_freq<15:0>	Ctcss/cdcss frequency setting Ctcss freq = ctcss_freq*2^16 khz It must be set to 134.4Hz when use standard cdcss mode When use ctcss/cdcss, this register must be set both

		in rx and tx state
4bH[7:0] 4cH[15:0]	4bH[7:0]=cdcss_code<23:16> 4cH[15:0]=cdcss_code<15::0>	Cdcss send/receive bit Note that MSB will be transmitted first!!! See 'RDA1846 register table' CDCSS MSB When use cdcss, this register must be set both in rx and tx state

23/24 bit CDCSS can controlled by 45H [4] (CDCSS_sel). CDCSS_sel=1 is 24 bit code ,=0 is 23bit code.

Such as TX 94.7Hz CTCSS :

4aH[15:0](ctcss_sentreg)=0.0974*(2^16) = 6383

Note: setting 45H [2:0]=000 when without subaudio

Add dcs_pos_det & dcs_neg_det register in 45H when use cdcss mode

9. SQ

Bit	Name	Function
30H[3]	sq_on	1 = on, then chip auto sq 0 = off
45H[3]	ctcss_sel	1 = ctcss_cmp/cdcss_cmp out via gpio 0 = ctcss/cdcss sdo out vio gpio
45H[10]	css_det_en	If 1, sq detection will add ctcss/cdcss detect result, then 1846 control 1846 voice output on or off.
48H[9:0]	th_h_sq<9:0> Sq open threshlod	Sq detect high th, rssi_cmp will be 1 when rssi>th_h_sq, unit 1/8dB 48H[9:3]＝Binary (135+ Sq open threshlod)
49H[9:0]	th_l_sq<9:0> Sq shut threshold	Sq detect low th, rssi_cmp will be 0 when rssi<th_l_sq && time delay meet, unit 1/8 dB 49H[9:3]＝Binary (135+ Sq shut threshlod)
54H[7]	sq_out_sel	If 1, the output gpio6 is sq & css_cmp; Else, the outputp gpio is sq only.

Auto SQ set 30H [3]=1(sq_on).

If auto SQ and subaudio detected at the same time,45H [10]=1 must be set.

48H[9:0] is Sq detect high th,49H[9:0] is Sq detect low th.

Such as Sq open threshold=-120dBm and Sq shut threshold=-122dBm

So 48H[9:3]= Binary (135+(-120))=0001111, 48H[9:0]= 0001111000

49H[9:3]= Binary (135+(-122))=0001101, 49H[9:0]= 0001101000

10. VOX

Bit	Name	Function
30H[4]	vox_on	1 = on, then chip auto vox 0 = off
41H[15:0]	th_h_vox<15:0> Vox open threshold	th_h_vox<15:0>=225* (open threshold) When vssi > th_h_vox, then vox will be 1 (unit mV)
42H[15:0]	th_l_vox<15:0> Vox Shut threshold	th_l_vox<15:0>=225* (shut threshold) When vssi < th_l_vox && time delay meet, then vox will be 0 (unit mV)

Such as vox open open threshold=2mV, vox shut threshold=2mV

So 42H[15:0]=225*1(mV)= Binary (225)= 0000000011100001

41H[15:0]=225*2(mV)= Binary (450)= 0000000111000010

11. Eliminating tail noise

While setting 30H [11]=1 eliminates tail noise when Tx and Rx, note turning on Tx and Rx CTCSS operation. Tx CTCSS phase can be controlled by 45H[15:14].

Bit	Name	Function
30H[11]	tail_elim_en	1 = tail elim enable 0 = disable
45H[15:14]	shift_select<1:0>	Select ctcss phase shift when use tail eliminating function when TX 00 = 120 degree shift 01 = 180 degree shift 10 = 240 degree shift 11 = reserved

12. DTMF

Bit	Name	Function
63H[15:10]	others<5:0>	000000
63H[9:8]	Dtmf_mode<1:0>	11 =transmit or receive Dtmf single tone2 01 =transmit or receive Dtmf dual tone1+tone2 others = disable

63H[7:4]	dtmf_time1<3:0>	Time interval for dual tone transmission Time = dtmf_time1*5ms
63H[3:0]	dtmf_time2<3:0>	Time interval for dtmf idle state Time = dtmf_time2*5ms
35H[15:0]	tone1_freq<15:0>	interval_v_reg= (Tone1 freq(kHz)* 2^12)
36H[15:0]	tone2_freq<15:0>	interval_c_reg= (Tone2 freq(kHz)* 2^12)
5cH[12]	dtmf_idle	Dtmf idle
66H[15:8]	dtmf_c0	697Hz 66H[15:8]= 01100001 12.8MHz and 25.6MHz 66H[15:8]= 01100001 13MHz and 26MHz
66H[7:0]	dtmf_c1	770Hz 66H[7:0]=01011011 12.8MHz and 25.6MHz 66H[7:0]=01011110 13MHz and 26MHz
67H[15:8]	dtmf_c2	852 Hz 67H[15:8]=01010011 12.8MHz and 25.6MHz 67H[15:8]= 01010111 13MHz and 26MHz
67H[7:0]	dtmf_c3	941 Hz 67H[7:0]=01001011 12.8MHz and 25.6MHz 67H[7:0]= 01001011 13MHz and 26MHz
68H[15:8]	dtmf_c4	1209 Hz 68H[15:8]=00101100 12.8MHz and 25.6MHz 68H[15:8]=00110001 13MHz and 26MHz
68H[7:0]	dtmf_c5	1336 Hz 68H[7:0]=00011110 12.8MHz and 25.6MHz 68H[7:0]=00011110 13MHz and 26MHz
69H[15:8]	dtmf_c6	1477 Hz 69H[15:8]=00001010 12.8MHz and 25.6MHz 69H[15:8]=00001111 13MHz and 26MHz
69H[7:0]	dtmf_c7	1633 Hz 69H[7:0]=11110110 12.8MHz and 25.6MHz 69H[7:0]=11111011 13MHz and 26MHz
6cH[10:5]	dtmf_index<5:0>	<5:3> : tone1 detect index <2:0> : tone2 detect index, will be used when single tone mode
6cH [4]	dtmf_flag	Dtmf code not valid flag 1 = not valid
6cH [3:0]	dtmf_code<3:0>	Dtmf code out Usually, F0~F7 is selected as 697, 770, 852, 941, 1209, 1336, 1477, 1633 Hz (default)

	F4	F5	F6	F7
F0	1	2	3	A
F1	4	5	6	B
F2	7	8	9	C
F3	E(*)	0	F(#)	D

TX and RX DTMF set 63H [8]=1(DTMF_en),close DTMF set 63H [8]=0.
Setting DTMF frequency 35H[15:0] (tone1_freq)（35H）and 36H[15:0] (tone2_freq) .Unite is 1/2^12KHz
Such as ：DTMF signal is 697Hz 和 1633Hz，
Tone1_freq<15:0> = round（0.697 *2^12）=2855
Tone2_freq<15:0> = round（1.633 *2^12）=6689
If tx single frequency signal, only setting tone2_freq and 63H [9:8]=11(single_tone), and 63H[7:4]=1111,63H[3:0]=0000.Or setting tone2_freq and 3CH[15:14]=01.
Rx DTMF:
Step1:set 66H,67H,68H,69H DTMF frequency according to reference clock
Step2: set DTMF_en=1 (63H[8]) if use INT mode, should set gpio2<1:0> to 01, and set int_grp_en<6> to 1
Step3 read dtmf_idle every 10ms until dtmf_idle=1 (5cH[12]) or wait INT when use INT mode
Step4: read dtmf_code<3:0> (6cH[3:0]
Step5: read dtmf_idle every 10ms until dtmf_idle=0 (5cH[12]) or write 00H=0x1846 (to clear INT) when use INT mode
Step6: jump to Step3
End of Rx DTMF, setting DTMF_en=0 and software jump out the circle Steps.
Tx DTMF:
Step1: setting DTMF sequence and the first DTMF frequency (ton1_freq and ton2_freq)
Step2: set DTMF_en=1 when needed (63H[8]) if use INT mode, should set gpio2<1:0> to 01, and set int_grp_en<6> to 1
Step3: read dtmf_idle every 10ms until dtmf_idle=1 (5cH[12]) or wait INT when use INT mode
Step4: setting the next DTMF frequency (ton1_freq and ton2_freq) according DTMF sequence
Step5: read dtmf_idle every 10ms until dtmf_idle=0 (5cH[12]) or write 00H, 0x1846 (to clear INT) when use INT mode
Step6: jump to Step3
End of Tx DTMF, setting DTMF_en=0 and software jump out the circle Steps.

13. Tx FM deviation

Bit	Name	Function
[15:13]	others	00
43H [12:6]	xmitter_dev<6:0>	Ctcss/cdcss + voice dev setting

43H [5:0]	c_dev<5:0>	Ctcss/cdcss dev setting

Adjusting 43H [12:6] (xmitter_dev) can change Tx FM deviation of voice and subaudio.
Adjusting 43H [5:0] (c_dev) can only change Tx FM deviation of CTCSS and CDCSS.

14. Rx voice range

Bit	Name	Function
44H[15:8]	others	00000000
44H[7:4]	volume1<3:0>	(0000)-15dB~(1111)0dB, step 1dB
44H[3:0]	volume2<3:0>	(0000)-15dB~(1111)0dB, step 1dB

Adjusting 44H [3:0] and 44H [7:4] can change Rx voice range.

15. TX and RX code

Set code mode:

Step1: set 58H[1:0]=11	set voice hpf bypass
Step2: set 58H[5:3]=111	set voice lpf bypass and pre/de-emph bypass
Step3 set 3CH[15:14]=10	set code mode
Step4: set 1FH[3:2]=01	set GPIO code in or code out

TX code mode：
Step1: 45H[2:0]＝010

RX code mode：
Step1: set 45H[2:0]=001
Step2: set 4dH[15:10]=000001

16. GPIO

Register 1fh.

Bit	Name	Function
15:14	gpio7<1:0>	00 =hi-z 01 = vox 10 = low 11 = high
13:12	gpio6<1:0>	00 =hi-z

		01 = sq, or =sq&ctcss/cdcss,when sq_out_sel=1 10 = low 11 = high
11:10	gpio5<1:0>	00 =hi-z 01 = txon_rf 10 = low 11 = high
9:8	gpio4<1:0>	00 =hi-z 01 = rxon_rf 10 = low 11 = high
7:6	gpio3<1:0>	00 =hi-z 01 = sdo 10 = low 11 = high
5:4	gpio2<1:0>	00 =hi-z 01 = int 10 = low 11 = high
3:2	gpio1<1:0>	00 =hi-z 01 = code_out/code_in 10 = low 11 = high
1:0	gpio0<1:0>	00 =hi-z 01 = css_out/css_in/css_cmp 10 = low 11 = high

17. INT

Register 2dh.

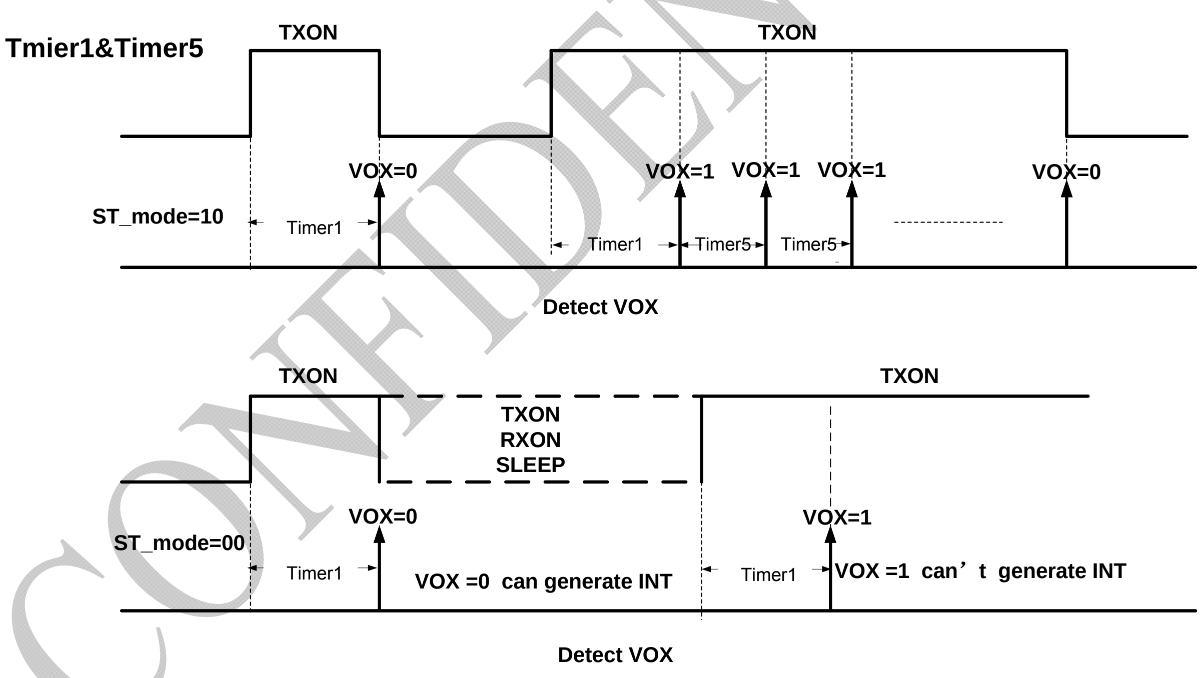
16' b0000_0000_0000

Bit	Name	Function
15:10	others <5:0>	000000
9:0	int_grp_en<9:0>	<9> :css_cmp_int enabl <8> : rxon_rf int enable <7> : txon_rf int enable <6> : dtmf_idle int enable

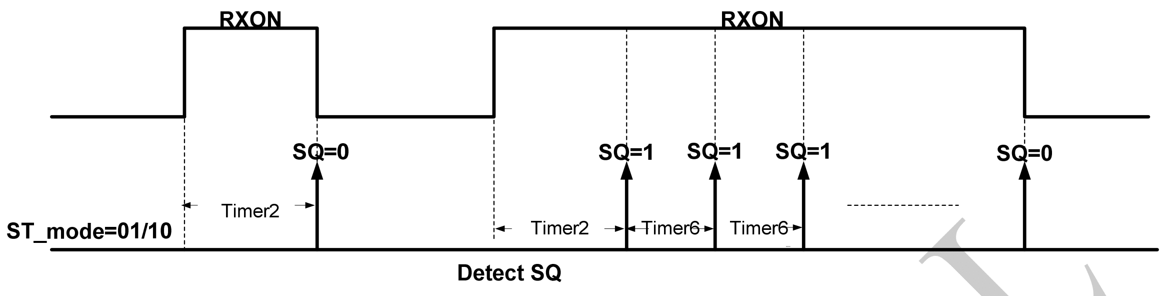
		<5> : ctcss phase shift detect int enable <4> : idle state time out int enable <3> : rxon_rf timerout int enable <2> : sq int enable; <1> : txon_rf time out int enable; <0> : vox int enable；

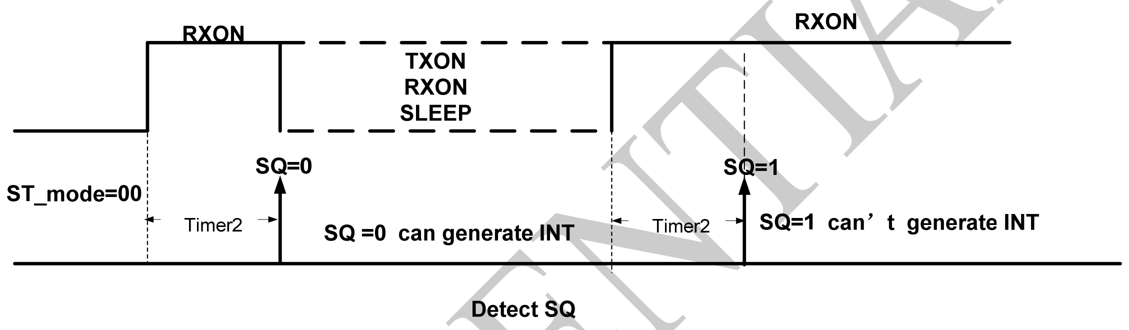
18. St_mode

Bit	Name	Function
30H[9:8]	st_mode<1:0>	11 = reserved 10 = txon_rf & rxon_rf auto 01 = rxon_rf auto, txon_rf manu 00 = txon_rf & rxon_rf manu

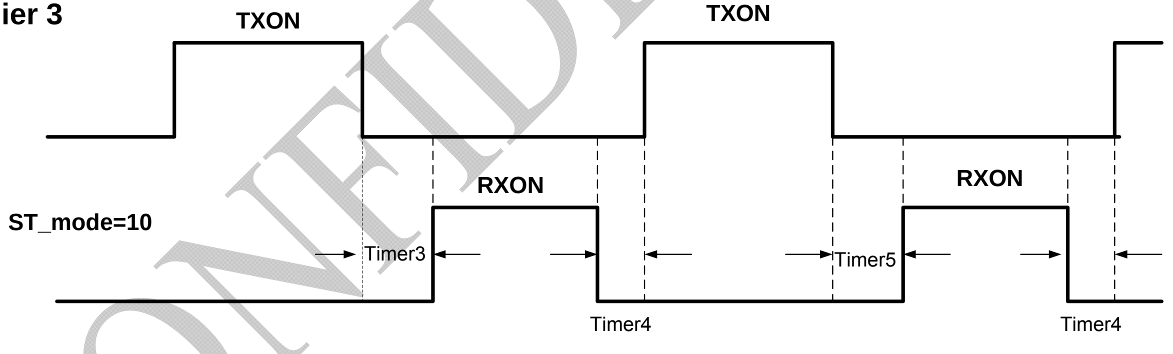


Tmier2 & Timer6

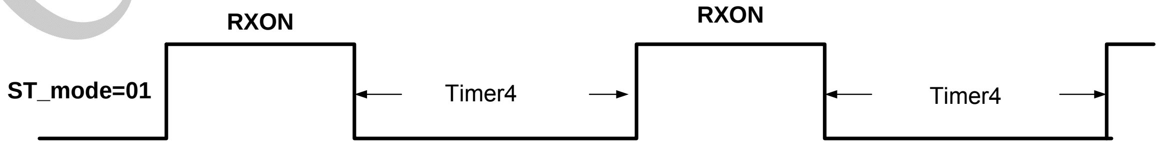


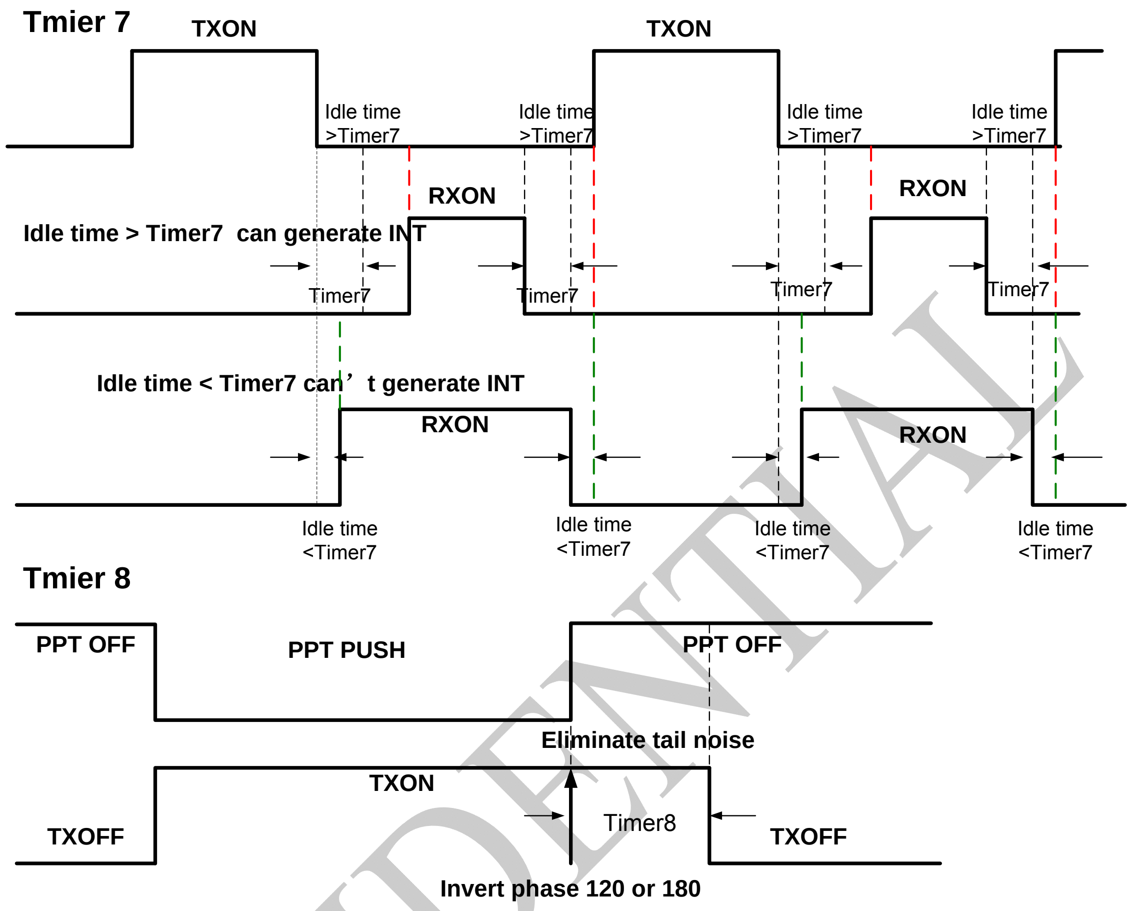


Tmier 3



Tmier 4





19. Pre-emphasis/De-emphasis filter

Bit	Name	Function
58H[3]	pre/de-emph	1=pre/de-emph bypass 0=normal

20. Only read register

Bit	Name	Function
5fH[9:0]	Rssi<9:0>	Received signal strength indication, unit 1/8dB
60H[14:0]	Vssi<14:0>	Voice signal strength indication, unit mV
6cH[10:5]	dtmf_index<5:0>	<5:3> : tone1 detect index <2:0> : tone2 detect index
6cH[3:0]	dtmf_code<3:0>	Dtmf code out

		1:f0+f4, 2:f0+f5, 3:f0+f6, A:f0+f7, 4:f1+f4, 5:f1+f5, 6:f1+f6, B:f1+f7, 7:f2+f4, 8:f2+f5, 9:f2+f6, C:f2+f7, E(*):f3+f4, 0:f3+f5, F(#):f3+f6, D:f3+f7

Such as：

Read 5fH[9:0]= Binary (110100000)=Dec(416)

So Received signal strength =(416*0.125)-135=(416/8)-135= -83dBm

21. Flag

Bit	Name	Function
5cH[12]	dtmf_idle	Dtmf idle
5cH [10]	rxon_rf	If 1, rxon is enable
5cH[9]	txon_rf	If 1, txon is enable
5cH[7]	invert_det	Ctcss phase shift detected
5cH [2]	css_cmp	Ctcss/cdcss compared
5cH [1]	SQ	Sq final signal out from dsp
5cH [0]	VOX	Vox out from dsp

22. Initial process

Refer to the ‘RDA18456 _register_table’

23. Register introduction

Register 30h.

Bit	Name	Function	Default
15:14	others	00	00
13:12	channel_mode	11 = 25khz channel mode 00 = 12.5khz channel mode 10,01=reserved	0
11	tail_elim_en	1 = tail elim enable 0 = disable	0
10	others	0	0
9:8	st_mode<1:0>	11 = reserved 10 = txon_rf & rxon_rf auto	00

		01 = rxon_rf auto, txon_rf manu 00 = txon_rf & rxon_rf manu	
7	mute	1 = mute when rxno 0 = no mute	0
6	tx_on	1 = on 0 = off	0
5	rx_on	1 = on 0 = off	0
4	vox_on	1 = on, then chip auto vox 0 = off	0
3	sq_on	1 = on, then chip auto sq 0 = off	0
2	pdn_reg	The same as pdn pin 1 = enable 0 = disable	0
1	chip_cal_en	1 = cal enable 0 = cal disable	0
0	soft_reset	1 = reset, then all the registers are reset to default value 0 = normal	0

Register 04h.

Bit	Name	Function	Default
15:1	others	0000_1111_0001_000	
0	clk_mode	12~14MHz:1 24~ 28MHz:0	1

Register 0ah.

Bit	Name	Function	Default
15:6	others	0000_0100_00	
5:0	pabias_voltage<5:0>	See TX Pa_bias output voltage	10_0000

Register 0fh.

Bit	Name	Function	Default
15:8	others		00000000
7:6	band_select	See setting RF band	00
5:0	others		100100

Register 29h.

Bit	Name	Function	Default
15:14	others		00
13:0	freq_reg	See setting frequency	0000000110010

Register 2ah.

Bit	Name	Function	Default

15:0	freq_reg	See setting frequency	0000010010110000

Register 2bh.

Bit	Name	Function	Default
15:0	xtal_freq	See reference clock	0011001000000000

Register 3ch.

Bit	Name	Function	Default
15:14	voice_sel<1:0>	See tx voice channel	00
13:0	others	00_1001_0101_1000	

Register 41h.

Bit	Name	Function	Default
15	others	0	
14:0	th_h_vox<14:0>	See vox	00_0000_0100_0000

Register 42h.

Bit	Name	Function	Default
15	others	0	
14:0	th_h_vox<14:0>	See vox	00_0000_0011_1100

Register 45h.

Bit	Name	Function	Default
15:14	shift_select<1:0>	See eliminating tail noise	00
13:12	others		00
11	Pos_det_en	See subaudio	1
10	css_det_en	See subaudio/sq	0
9:8	others		10
7	neg_det_en	See subaudio	1
6:5	others		00
4	cdcss_sel	See subaudio	0
3	others		0
2:0	c_mode<2:0>	See subaudio	000

Register 48h.

Bit	Name	Function	Default
15:10	others		000000
9:0	Sq open threshold	See SQ	0001010000

Register 49h.

Bit	Name	Function	Default
15:10	others		000000
9:0	Sq shut threshold	See SQ	0000111100

Register 4ah.

Bit	Name	Function	Default

15:0	ctcss_freq	See Subaudio	0001100110011001

Register 4bh.

Bit	Name	Function	Default
15:8	others	Read as zeros	0000_0000
7:0	cdcss_code	See subaudio	0110_0101

Register 4ch.

Bit	Name	Function	Default
15:0	cdcss_code	See subaudio	1101_1000_0001_0110

Register 54h.

Bit	Name	Function	Default
15:13	others	0001_0001	
7	sq_out_sel	See sq	0
6:0	others	100_1000	

Register 63h.

Bit	Name	Function	Default
15:10	Reserved<5:0>	000000	0000
9	single_tone	See dtmf	0
8	dtmf_en	See dtmf	0
7:4	dtmf_time1<3:0>	See dtmf	1000
3:0	dtmf_time2<3:0>	See dtmf	1000

Register 66h

Bit	Name	Function	Default
15:8	dtmf_c0	697Hz	0110_0001
7:0	dtmf_c1	770Hz	0101_1011

Register 67h.

Bit	Name	Function	Default
15:8	dtmf_c2<7:0>	852Hz	0101_0011
7:0	dtmf_c3<7:0>	941Hz	0100_1011

Register 68h.

Bit	Name	Function	Default
15:8	dtmf_c4<7:0>	1209Hz	0010_1100
7:0	dtmf_c5<7:0>	1336Hz	0001_1110

Register 69h.

Bit	Name	Function	Default
15:8	dtmf_c6<7:0>	1477Hz	0000_1010
7:0	dtmf_c7<7:0>	1633Hz	1111_0110

Change List

Rev	Date	Author	Change Description
0.1	2009-5-20	Liu Ge & Liu ya nan	Original draft
1.1	2009-6-17	Liu Ge & Liu ya nan	
1.1	2009-10-13	Liu Ge	Add register indroduction
1.2	2009-11-13	Liu Ge	Modify DTMF and RSSI indroduction

Disclaimer

For technical questions and additional information about RDA Microelectronics Inc.:

Website: www.rdamicro.com
Mailbox: info@rdamicro.com

RDA Microelectronics (Shanghai), Inc.	RDA Microelectronics (Beijing), Inc.
Tel: +86-21-50271108	Tel: +86-10-63635360
Fax: +86-21-50271099	Fax: +86-10-82612663