RF Point Pte Ltd
RDA Exclusive China Overseas Sales Agent

RDA16112 Programming Guide

Change List

Rev	Date	Author	Change Description
1.0	2015-5-9	Xiaofei Peng	Original draft

1. Introduction

RDA16112 enable software programming through I2C interface. Software controls chip working states, and allows user read status registers to get operation result through I2C interface.

2. I2C Interface

The I2C interface of RDA16112 is compliant to I2C-Bus Specification 2.1(Fast-mode, bit rate up to 400Kbit/s). It includes two pins: SCLK and SDA. SCLK is an input pin; SDA is a bi-direction pin.

The 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 device address {5'b00011, MA1, MA0} and an r/$\overline{w}$ bit. (MA1 and MA0's values depends on input pin ADD's voltage, default 2'b00.) The ACK (or NACK) is always sent out by the receiver. When in write transfer process, data bytes are written out from MCU, and when in read transfer process, data bytes are read out from RDA16112.

The RDA16112 contains status/control registers. These read/write registers are addressed as sub-address on the I2C bus. RDA16112 I2C interface supports both single and sequential register access. Software could follow the following ways to perform register read/write access:

Random access single write

Start	Device address	W	A	Register address	A	Register data	A	stop

Random access sequential write

Start	Device address	W	A	Register N address	A	Register N Data	A	Register N+1Data	…	Register N+M Data	A	stop

Random access single read

Start	Device address	W	A	Register address	A	start	Device address	R	A	Register Data	N	stop

Random access sequential read

Start	Device address	W	A	Register N address	A	start	Device address	R	A	Register N Data	A	Register N+1 Data	A	...	Register N+M Data	N	stop

W: Write Bit (0: write; 1: read)
A: Acknowledge Bit (ACK)

N: Not Acknowledge Bit (NO ACK)

For random access single write transfer, MCU sends out the START signal, RDA16112's device address and 1 bit write signal in sequence to the I2C bus. After receiving RDA16112' s ACK signal, MCU sends out the target register's address (8 bits) to RDA16112 and then programs this register with proper data (8 bits). A STOP signal is sent out by MCU to end this transfer when programming is finished.

For random access sequential write transfer, MCU sends out the START signal, RDA16112's device address and 1 bit write signal to the I2C bus. After receiving RDA16112's ACK signal, MCU sends the sequential registers' first address (8 bits) to RDA16112. Then MCU could program these registers sequentially. A STOP signal is sent out by MCU to end this transfer when programming is finished.

For random access single read transfer, MCU first sends out the START signal, the RDA16112's device address and 1 bit write signal to the I2C bus. After receiving RDA16112's ACK signal, MCU sends the target register's address (8 bits) to the interface. Then MCU should send another command byte, including a RESTART signal, the RDA16112's device address and 1 bit read signal. Then RDA16112 will send the register's data to MCU through I2C bus. After the byte has been received, MCU should send a NO ACK response signal and a STOP signal to finish this read transfer.

For random access sequential write transfer, MCU first sends out the START signal, the RDA16112's device address and 1 bit write signal to the I2C bus. After receiving RDA16112's ACK signal, MCU sends the sequential registers' first address (8 bits) to RDA16112. Then MCU should send another command byte to the interface, including a RESTART signal, the RDA16112's device address and 1 bit read signal. Then RDA16112 starts to transfer the sequential registers' data byte by byte to MCU through I2C bus, each byte followed by an ACK response signal generated by the MCU. After all target registers' data have been read, MCU should send a NO ACK signal and a STOP signal to finish this read transfer.

3. Frequency Setting

When frequency is changed in reset&config state, software need program correct channel frequency. The frequency value format used in RDA16112 is a 30-bit binary format, put in register 07H [5:0], 08H[7:0], 09H[7:0] and 0aH[7:0]. Software need to translate the channel frequency into 30-bit binary format, and then write into RDA16112.

Equation of 30-bit binary format and channel frequency :(f is frequency, {07H [5:0], 08H [7:0], 09H [7:0], 0aH [7:0]} is the 30-bit format frequency value, f_xtal is the xtal frequency.)

{07H[5:0], 08H[7:0],09H[7:0],0aH[7:0]} = dec2bin(f * (2^21) / f_xtal)

For example: If the channel frequency is 950MHz, xtal frequency is 27M, translate the 950MHz into 30-bit binary format:

dec2bin(950 * (2^21) /27) = **30'b0000_0100_0110_0101_1110_1101_0000_1001**

software should program RDA16112 register 07H,08H,09H and 0AH using the follow programming sequence:

Mov 8'b0000_0100(04), 07H
Mov 8'b0110_0101(65), 08H
Mov 8'b1110_1101(ed), 09H
Mov 8'b0000_1001(09), 0aH

Attention: Frequency changing should always be done in reset&initial state or idle state. When chip in normal working state, programmer should write 1'b0 to rxon register (04h [1]) before setting new frequency value. After frequency setting has been finished, programmer could write 1 to rxon register (04h [1]), leading the chip into self calibration state and then into receiving state again.

4. Filter bandwidth setting

The filter bandwidth of RDA16112 is adjustable (from 4M to 40M, step 1M). Software could control RDA16112's filter bandwidth by setting register 0bh[5:0]. For example, if the target bandwidth is 10MHz, software should program 0bh [5:0] to 6'd10; if the target bandwidth is 22MHz, software should program 0bh [5:0] to 6'd22, etc...

5. Register configuration sequence

Programmer should initialize and configure RDA16112's register file following the sequence shown in Figure 3, and a detailed register configuration example is also provided in this section.

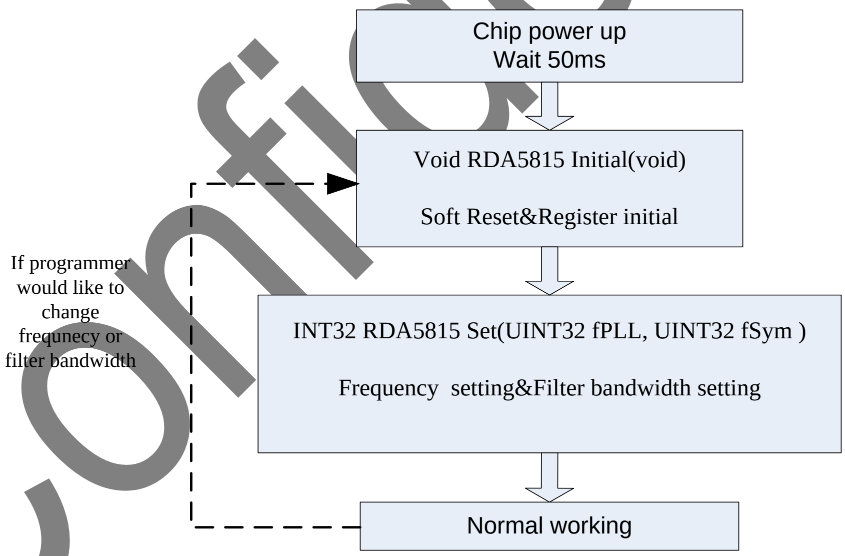

Figure 3 RDA5815 configuration sequence

void RDA5815 Initial(void)

This operation include the soft reset,setting pll,setting agc,etc.

INT32 RDA5815 Set(UINT32 fPLL, UINT32 fSym)

This operation is setting the frequency and filter bandwidth. The unit of frequency is MHz and unit of bandwidth is KHz.

6. Register summary

Register 00h Chip ID_1 (read only) default: 8'b0101_1000

Bit	Name	Function	Note
7:0	Chip_ID[11:4]	5812 chip id Default 8'b0101_1000	read only

Register 01h Chip_ID_2 (read only) default: 8'b1100_0000

Bit	Name	Function	Note
7:4	Chip_ID[3:0]	Default: 4'b1100	read only
3:0	Revision_ID	Default: 4'b0000	read only

Register 02h Device ID (read only) default: 8'b0000_1100

Bit	Name	Function	Note
7	Reversed	1'b0	
6:2	Device_ID_fix[4:0]	Chip address Fixed Default 5'h0_0011	read only
1	MA1	address 1 selected via pin ADD configuration Default 0	read only
0	MA0	address 1 selected via pin ADD configuration Default 0	read only

Register 04h Config Control register default: 8'b0000_0101

Bit	Name	Function	Note
7	enable	if 1,enable interface; if 0,sleep mode, only bandgap,xtal and adr_encoder module power on Default: 1'b0	
6	Bypass_rxon_agc_enable	Default: 1'b0	
5	rf_bypass_enable_b	0: power up 1: power down Default: 1'b0 The bypass buf functions at power-up of the chip by default. If this function is not required, set this bit to	

		logic ‘1’, which will save about 6mA current.	
4	bypass_tuning_enable	If set to 1,bypass tuning calibration when changing the filter bandwdith; Default: 1’b0	
3	bypass_rxon_dc_cal_enable	If set to 1,bypass dc offset calibration when each time changing the working freq or the filter bandwdith; Default: 1’b0	
2	agc_en_b	If 0,AGC enable; If 1,disable AGC; Default 1	
1	rxon	If 1 , in rxon mode Default 0	
0	Soft_resetn	If 0, then reset all the register of chip Default 1	

Register 07h PLL frequency setting1 (950MHZ) default: 8’b0000_0100

Bit	Name	Function	Note
7:0	Freq_synthesize_reg [31:24]	pll freq setting , from MSB to LSB :10 integer bits,22 decimal [7:2] interger bits	

Register 08h PLL frequency setting 2 default: 8’b0110_0101

Bit	Name	Function	Note
7:6	Freq_synthesize_reg [23:22]	[1:0] integer bits	
5:0	Freq_synthesize_reg [21:16]	pll freq setting , 22 decimal bits [21:16]. decimal bits[21:16].	

Register 09h PLL frequency setting 3 default:8’b1110_1101

Bit	Name	Function	Note
7:0	Freq_synthesize_reg [15:8]	pll freq setting , 22 decimal bits[15:8]. decimal bits[15:8].	

Register 0ah PLL frequency setting 4 default:8’b0000_1001

Bit	Name	Function	Note
7:0	Freq_synthesize_reg [7:0]	pll freq setting , 22 decimal bits[7:0]. decimal bits[7:0].	

Register 0bh filter bandwidth setting register default:8’b0000_1010

Bit	Name	Function	Note
7	filter_bw_control_bit_extra	Default 1’b0 when the filter bandwidth between 4M~8M, we use this bit to make the BW precision to 0.5M. eg: when BW=4M,we could set filter_bw_control_bit 6’b00_0100,and filter_bw_control_bit_add 1’b0; when BW 4.5M,we could set filter_bw_control_bit 6’b00_0100,and filter_bw_control_bit_add 1’b1;	
6	filter_fc_bit_dr	If 1,filter fc control bits using value direct decoded from filter_bw_control_bit; If 0, filter fc control bits using tuning calibration result. default: 0	
5:0	filter_bw_control_bit [5:0]	filter bandwidth setting(4MHz ~ 40MHz) eg:if user need bandwdth 10M, set this register to 6’b00_1010.	

Disclaimer

The information provided here is believed to be reliable; RDA Microelectronics assumes no reliability for inaccuracies and omissions. RDA Microelectronics assumes no reliability for the use of this information and all such information should entirely be at the user's own risk. Specifications described and contained here are subjected to change without notice on the purpose of improving the design and performance. All of this information described herein should not be implied or granted for any third party. RDA Microelectronics does not authorize or warrant any RDA products for use in the life support devices or systems.

For technical questions and additional information about RDA Microelectronics Inc.:
Website: www.rdamicro.com or www.rf-point.com.sg
Mailbox: info@rdamicro.com or enquiry@rf-point.com.sg

RDA Microelectronics (Shanghai), Inc.
Tel: +86-21-50271098
Fax: +86-21-50271099

RDA Microelectronics (Beijing), Inc.
Tel: +86-10-62635360
Fax: +86-10-82612663