SI4463通用板参数设置

该文档用于介绍SI4463各个参数的计算方法

void InitRfTestBand()

{

unsigned char inter=0x00; //标记第几次进入中断

unsigned char INT\_STA[8];

unsigned char res;

unsigned char PART\_FUNC[8];

unsigned char GPIO\_STATE[7]; //接受GPIO功能的相关返回数据

unsigned char pkt\_config[3];

unsigned char DEVICE\_STATE[2];

unsigned char FRR\_READ[4];

unsigned char GLOBAL[3];

unsigned char MODEM;

unsigned char SYNTH\_PAR[7];

unsigned char INT\_0=0x00;

unsigned char INT\_1=0x00;

unsigned char INT\_2=0x00;

unsigned char INT\_3=0x00;

unsigned char PRE\_1=0X00;

unsigned char PRE\_2=0X00;

unsigned char FR0=0x00;

unsigned char FR1=0x00;

unsigned char FR2=0x00;

unsigned char FR3=0x00;

unsigned char FR4=0X00;

unsigned char FR5=0X00;

unsigned char FR6=0X00;

void NIRQ\_CONFIG(void);

unsigned char clickcount=0;

unsigned char datalength=0x07;

unsigned char receivebuffer[7];

unsigned char receiveflag=0;//接收数据标志

unsigned char data\_fifo[10]={0x55,0x54,0x53,0x52,0x51,0x50,0xAA,0xAA,0xAA,0xAA};

unsigned char CMD[16];

unsigned char incount=0;

unsigned char API\_WRITER[12];

// API\_POWER\_UP();

//power up

API\_WRITER[0]=CMD\_POWER\_UP; API\_WRITER[1]=0x01;

API\_WRITER[2]=0x00; API\_WRITER[3]=0x01;

API\_WRITER[4]=0xc9; API\_WRITER[5]=0xc3;

API\_WRITER[6]=0x80;

API\_SendCommand(0x07,API\_WRITER);

for(int m=0;m<100;m++)

{

for(int n=0;n<100;n++)

{

;

}

}

// API\_READ\_INT\_STATUS(0x08,INT\_STA); //清除NIRQ管脚的中断标志

API\_Get\_All\_IntStatus(0x08,INT\_STA);

P1IFG=0;

//API\_REQUEST\_DEVICE\_STATE(DEVICE\_STATE); //现在处于READY模式 ===读取正常

/\*===================================

COMMAND return PART\_Information 0X01

=====================================\*/

API\_PART\_INFO(0x08, PART\_FUNC);

/\*===================================

COMMAND GPIO\_CONFIG 0X13

=====================================\*/

//API\_GPIO\_CFG(0x07,GPIO\_STATE);

API\_WRITER[0]=CMD\_GPIO\_PIN\_CFG; API\_WRITER[1]=0x21;

API\_WRITER[2]=0x20; API\_WRITER[3]=0x20;

API\_WRITER[4]=0x21; API\_WRITER[5]=0x00;

API\_WRITER[6]=0x00; API\_WRITER[7]=0x00;

API\_SendCommand(0x08,API\_WRITER);

API\_WaitforCTS();

/\*===================================

GLOBAL CONTROL 0X0000/01/03

=====================================\*/

GLOBAL\_XO\_TUNE(0x50); //原先是0x52 调制电容

//GLOBAL\_CLK\_CFG(0x02);

// GLOBAL\_CONFIG(0x60); //原来写入的是0x00

// GLOBAL\_CONFIG(0x00); //原来写入的是0x00

//433发送数据包测试代码

GLOBAL\_CONFIG(0x60);

/\*返回写入的数据，验证是否正确写入\*/

// GLOBAL[0]=GET\_GLOBAL\_XO\_TUNE();

//GLOBAL[1]=GET\_GLOBAL\_CLK\_CFG();

// GLOBAL[2]=GET\_GLOBAL\_CONFIG();

/\*===================================

INT\_ENABLE的相关设置 0X01/00-0X01/03

需要在进行重新设定

=====================================\*/

INT\_CTL\_ENABLE(0x01); //0x01

INT\_CTL\_PH\_ENABLE(0x18); //0x20

//INT\_CTL\_MODEM\_ENABLE(0x00);

//INT\_CTL\_CHIP\_ENABLE(0x00);

/\*===================================

FRR的相关设置 0X02/00-0X02/03

=====================================\*/

FRR\_CTL\_A\_MODE(0x01); //0X0200

FRR\_CTL\_B\_MODE(0x03); //0X0201

FRR\_CTL\_C\_MODE(0x05); //0X0202

FRR\_CTL\_D\_MODE(0x09); //0X0203

/\*

在MODEM调制的相关寄存器中，会发现下标并不是很完整，

参考html格式的API寄存器的相关介绍就会查找到

相关不足的寄存器

\*\*注意\*\*

在参考WDS生成的寄存器的值的时候，会发现

GROUP:0X20中并没有ID:0X0D-0X17的相关设置，

在html格式的API文档中会有介绍

0X0D-0X0E:MODEM\_FREQ\_OFFSET

0X0F-0X17：MODEM\_TX\_FILTER\_COEFF\_(8\_0)

\*/

/\*===================================

MODEM CONTROL GROUP:0X2000

=====================================\*/

/\* CMD[0]=0X02; CMD[1]=0X00; CMD[2]=0X07;

CMD[3]=0X01; CMD[4]=0X86; CMD[5]=0XA0;

CMD[6]=0X01; CMD[7]=0XC9; CMD[8]=0XC3;

CMD[9]=0X80; CMD[10]=0X00; CMD[11]=0X05;

API\_SET\_PROPERTY\_X(0X20,0X0C,0X00,CMD);

RF\_MODEM\_FREQ\_DEV\_0(0X76);

MODEM=GET\_MODEM\_CLKING\_BAND();\*/

/\*

数据调制类型的确定

\*/

// MODEM\_MOD\_TYPE(0X08); //载波模式

MODEM\_MOD\_TYPE(0X03); //2GFSK调制模式

MODEM\_MAP\_CONTROL(0X00);// control bit mapping

/\*

该寄存器很重要，在后面的改变频偏中

用到了该寄存器的值

应用到

1：频偏寄存器计算

2：跳频寄存器计算

\*/

// MODEM\_CLKGEN\_BAND(0X08);

MODEM\_CLKGEN\_BAND(0X0a);

/\*

与发送数据速率有关的寄存器：

MODEM\_DATA\_RATE\_2;

MODEM\_DATA\_RATE\_1

MODEM\_DATA\_RATE\_0

TX\_NCO\_MODE\_3

TX\_NCO\_MODE\_2

TX\_NCO\_MODE\_1

TX\_NCO\_MODE\_0

计算公式：

NCO\_CLK\_FREQ=(MODEM\_DATA\_RATE\*Fxtal\_hz)/MODEM\_TX\_NCO\_MODEM

TX\_DATA\_RATE=(NCO\_CLK\_FREQ)/TXOSR;

TXOSR为TX\_NC0\_MODE寄存器中的一位

\*/

/\*

MODEM\_DATA\_RATE\_2(0X00);

MODEM\_DATA\_RATE\_1(0X4E);

MODEM\_DATA\_RATE\_0(0X20);

MODEM\_TX\_NCO\_MODE\_3(0X04);

MODEM\_TX\_NCO\_MODE\_2(0X2D);

MODEM\_TX\_NCO\_MODE\_1(0XC6);

MODEM\_TX\_NCO\_MODE\_0(0XC0);

\*/

//433 发送数据包测试代码，数据率10kbps

MODEM\_DATA\_RATE\_2(0X06);

MODEM\_DATA\_RATE\_1(0X1a);

MODEM\_DATA\_RATE\_0(0X80);

MODEM\_TX\_NCO\_MODE\_3(0X05);

MODEM\_TX\_NCO\_MODE\_2(0Xc9);

MODEM\_TX\_NCO\_MODE\_1(0Xc3);

MODEM\_TX\_NCO\_MODE\_0(0X80);

/\*

与频偏有关的寄存器的设置：

MODEM\_FREQ\_DEV\_2

MODEM\_FREQ\_DEV\_1

MODEM\_FREQ\_DEV\_0

计算公式：

MODEM\_FREQ\_DEV=(2^19\*Desirzed\_Dev\_Hz)/Npresc\*freq\_Xo;

Npresc参考MODEM\_CLKEN\_BAND

参考该寄存器里面额SY\_SEL位，确定是除4还是除2

\*\*注意\*\*

所以在设置频偏的时候该寄存器也是很重要的一个寄存器

\*/

/\*

MODEM\_FREQ\_DEV\_2(0X00);

MODEM\_FREQ\_DEV\_1(0X00);

MODEM\_FREQ\_DEV\_0(0X38);\*/

//频偏设置：20KHz

//433 数据包发送测试代码

MODEM\_FREQ\_DEV\_2(0X00);

MODEM\_FREQ\_DEV\_1(0X05);

MODEM\_FREQ\_DEV\_0(0X76);

/\*

MODEM\_TX\_RAMP\_DELAY:

summary:TX ramp\_down\_delya setting

参考WDS生成的值进行设置

\*/

// MODEM\_TX\_RAMP\_DELAY(0X03);

//发送延时

MODEM\_TX\_RAMP\_DELAY(0X01);

//RF\_MODEM\_TX\_RAMP\_DELAY\_6

CMD[0]=0x08;CMD[1]=0x03;CMD[2]=0x80;CMD[3]=0x00;

CMD[4]=0x20;CMD[5]=0x20;CMD[6]=0x20;

API\_SET\_PROPERTY\_X(0x20,0x06,0x1a,CMD);

/\*

API\_WRITER[0]=0X01; API\_WRITER[1]=0x00;

API\_WRITER[2]=0x08; API\_WRITER[3]=0x03;

API\_WRITER[4]=0x80; API\_WRITER[5]=0x00;

API\_WRITER[6]=0x20; API\_WRITER[7]=0x20;

API\_SET\_PROPERTY\_X(Group\_MODEM,0X08,ad\_MODEM\_TX\_RAMP\_DELAY,API\_WRITER);

\*/

//RF\_MODEM\_BCR\_OSR\_1\_9

CMD[0]=0x01;CMD[1]=0x77;CMD[2]=0x01;CMD[3]=0x5D;

CMD[4]=0x86;CMD[5]=0x00;CMD[6]=0xAF;CMD[7]=0x02;

CMD[8]=0xC2;

API\_SET\_PROPERTY\_X(0x20,0x09,0x22,CMD);

/\*

API\_WRITER[0]=0X00; API\_WRITER[1]=0x5E;

API\_WRITER[2]=0x05; API\_WRITER[3]=0x76;

API\_WRITER[4]=0x1A; API\_WRITER[5]=0x05;

API\_WRITER[6]=0x72; API\_WRITER[7]=0x02;

API\_WRITER[8]=0X00;

API\_SET\_PROPERTY\_X(Group\_MODEM,0X09,ad\_MODEM\_TX\_RAMP\_DELAY,API\_WRITER);

\*/

//RF\_MODEM\_AFC\_GEAR\_7

CMD[0]=0x04;CMD[1]=0x36;CMD[2]=0x80;CMD[3]=0x1D;

CMD[4]=0x10;CMD[5]=0x04;CMD[6]=0x80;

API\_SET\_PROPERTY\_X(0x20,0x07,0x2C,CMD);

//RF\_MODEM\_AGC\_CONTROL\_1

CMD[0]=0xE2;

API\_SET\_PROPERTY\_X(0x20,0x01,0x35,CMD);

//RF\_MODEM\_AGC\_WINDOW\_SIZE\_9

CMD[0]=0x11;CMD[1]=0x52;CMD[2]=0x52;CMD[3]=0x00;

CMD[4]=0x1A;CMD[5]=0xFF;CMD[6]=0xFF;CMD[7]=0x00;

CMD[8]=0x2A;

API\_SET\_PROPERTY\_X(0x20,0x09,0x38,CMD);

//RF\_MODEM\_OOK\_CNT1\_8

CMD[0]=0xA4;CMD[1]=0x02;CMD[2]=0xD6;CMD[3]=0x83;

CMD[4]=0x00;CMD[5]=0xAD;CMD[6]=0x01;CMD[7]=0x80;

API\_SET\_PROPERTY\_X(0x20,0x08,0x42,CMD);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*有此往下为GROUP:0X21的相关设置，PDF数据手册上没有给出介绍\*/

CMD[0]=0xa2;CMD[1]=0x81;CMD[2]=0x26;CMD[3]=0xaf;

CMD[4]=0x3f;CMD[5]=0xee;CMD[6]=0xc8;CMD[7]=0xc7;

CMD[8]=0xdb;CMD[9]=0xf2;CMD[10]=0x02;CMD[11]=0x08;

API\_SET\_PROPERTY\_X(0x21,0x0c,0x00,CMD);

CMD[0]=0x07;CMD[1]=0x03;CMD[2]=0x15;CMD[3]=0xfc;

CMD[4]=0x0f;CMD[5]=0x00;CMD[6]=0xa2;CMD[7]=0x81;

CMD[8]=0x26;CMD[9]=0xaf;CMD[10]=0x3f;CMD[11]=0xee;

API\_SET\_PROPERTY\_X(0x21,0x0c,0x0c,CMD);

CMD[0]=0xc8;CMD[1]=0xc7;CMD[2]=0xdb;CMD[3]=0xf2;

CMD[4]=0x02;CMD[5]=0x08;CMD[6]=0x07;CMD[7]=0x03;

CMD[8]=0x15;CMD[9]=0xfc;CMD[10]=0x0f;CMD[11]=0x00;

API\_SET\_PROPERTY\_X(0x21,0x0c,0x18,CMD);

/\*有此往上为GROUP:0X21的相关设置，PDF数据手册上没有给出介绍\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*===================================

FREQUENCY的相关设置 GROUP:0X4000

FREQ INTE OR FRAC CONTROL

=====================================\*/

/\*CMD[0]=0X3D; CMD[1]=0X0D; CMD[2]=0X55;

CMD[3]=0X55; CMD[4]=0X44; CMD[5]=0X44;

CMD[6]=0X20; CMD[7]=0XFE;

API\_SET\_PROPERTY\_X(0X40,0X08,0X00,CMD);\*/

/\*

载波频率的设定：

FREQ\_CONTROL\_INTE

FREQ\_CONTROL\_FRAC\_2

FREQ\_CONTROL\_FRAC\_1

FREQ\_CONTROL\_FRQC\_0

//这两个寄存器用于跳频步长的设定

FREQ\_CONTROL\_STEP\_SIZE\_1

FREQ\_CONTROL\_STEP\_SIZE\_0

FREQ\_CONTROL\_W\_SIZE

中心频率计算公式：

RF\_channel=(fc\_inte+(fc\_frac/2^19))\*(2\*freq\_xo)/outdiv

\*/

/\*

跳频步长的计算公式：

FREQ\_CTRL\_CHAN\_STEP\_SIZE=(2^19\*outdiv\*Desired\_step\_size)/(Npresc\*freq\_X0);

Npresc参考MODEM\_CLKEN\_BAND

参考该寄存器里面额SY\_SEL位，确定是除4还是除2

\*\*注意\*\*

所以在设置频偏的时候该寄存器也是很重要的一个寄存器

\*/

/\* FREQ\_CONTROL\_INTE(0X3C);

FREQ\_CONTROL\_FRAC\_2(0X0A);

FREQ\_CONTROL\_FRAC\_1(0XAA);

FREQ\_CONTROL\_FRAC\_0(0XAA);

FREQ\_CONTROL\_W\_SIZE(0X20);

\*/

FREQ\_CONTROL\_INTE(0X38);

FREQ\_CONTROL\_FRAC\_2(0X0d);

FREQ\_CONTROL\_FRAC\_1(0Xdd);

FREQ\_CONTROL\_FRAC\_0(0Xdd);

FREQ\_CONTROL\_CHANNEL\_STEP\_SIZE\_1(0x44);//跳频步长 250KHz

FREQ\_CONTROL\_CHANNEL\_STEP\_SIZE\_0(0x44);

FREQ\_CONTROL\_VCOCNT\_RX\_ADJ(0xfe);

FREQ\_CONTROL\_W\_SIZE(0X20);

// FREQ\_CONTROL\_INTE(0x3d); //中心频点470.1MHz 步长：200khz

//FREQ\_CONTROL\_FRAC\_2(0x0d);

// FREQ\_CONTROL\_FRAC\_1(0x70);

// FREQ\_CONTROL\_FRAC\_0(0xa3);

// FREQ\_CONTROL\_CHANNEL\_STEP\_SIZE\_1(0x36);

// FREQ\_CONTROL\_CHANNEL\_STEP\_SIZE\_0(0x9d);

// FREQ\_CONTROL\_W\_SIZE(0x20);

// FREQ\_CONTROL\_VCOCNT\_RX\_ADJ(0xfe);

/\*===================================

PA CONTROL GROUP:0X2200

=====================================\*/

/\*

与PA有关的寄存器的设置：

PA\_MODE:

PA\_PWR\_LVL:

PA\_BIAS\_CLKDUTY:

PA\_TC:

\*/

PA\_MODE( 0x08 );

PA\_PWR\_LVL(0x7F );

PA\_BIAS\_CLKDUTY( 0x00 );

PA\_TC( 0x3D );

/\*===================================

PREAMBLE的相关设置 0X1000/04

//前导码发送8字节，标准前导码 1010，字节组，不采用曼彻斯特编码

=====================================\*/

PREAMBLE\_TX\_LENGTH(0x0A); //0X08原先写入的值

PREAMBLE\_CONFIG(0x31);

/\*===================================

SYNC的相关设置 0X1100 0X1101 0X1102 0X1103 0X1104

发送两个字节的同步字，不采用4FSK和曼彻斯特编码,发送的同步字为2D,D4

=====================================\*/

SYNC\_CONFIG(0x01); //0X1100

SYNC\_BITS\_31\_24( 0xB4); //0X1101

SYNC\_BITS\_23\_16( 0x2B); //0X1102

/\*===================================

PKT的相关设置 GROUP:0X1200

数据包字段1的相关控制

=====================================\*/

PKT\_CONFIG1(0x02); //0x1206

PKT\_TX\_THRESHOLD(0X3A); //0x120b

PKT\_FIELD\_1\_LENGTH\_12\_8(0x00); //0x12d

PKT\_FIELD\_1\_LENGTH\_7\_0(datalength); //0x120e

PKT\_FIELD\_1\_CONFIG(0x04); //0x120f

/\*==============================================

GROUP:0X23 SYNTH\_PFDCP\_CPFF

0x2C, 0x0E, 0x0B, 0x04, 0x0C, 0x73, 0x03

==============================================\*/

SYNTH\_PFDCP\_CPFF(0X2C);

SYNTH\_PFDCP\_CPINT(0X0E);

VCO\_KV(0X0B);

SYNTH\_LPFILT3(0X04);

SYNTH\_LPFILT2(0X0C);

SYNTH\_LPFILT1(0X73);

SYNTH\_LPFILT0(0X03);

}

该处代码介绍了每个参数的计算方法，给出了与该参数相关的寄存器。