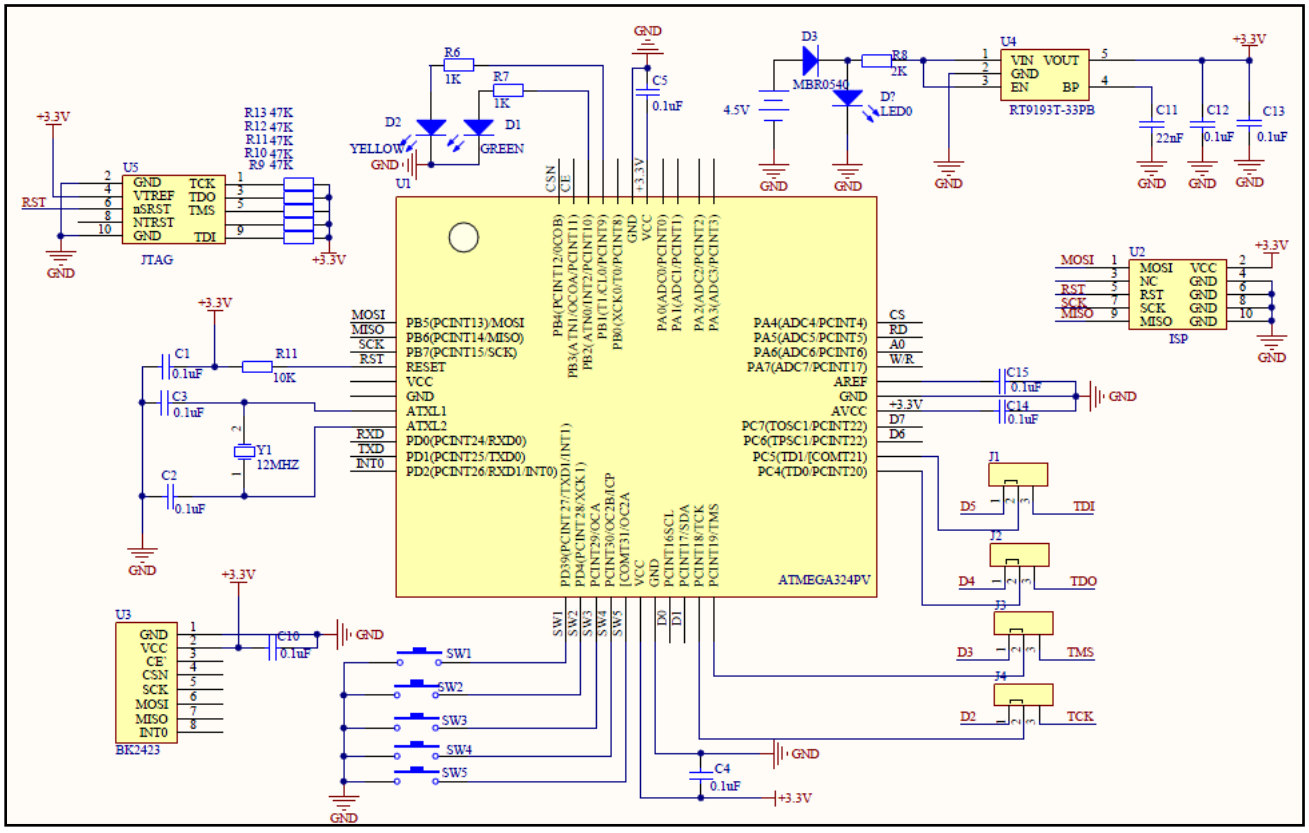
AN0008- RF-2423 Communication In 250Kbps Air Rate

1. General Description

This document describe how BK2423 achieve long-distance of communication in 250Kbps air rate on DEMO board。The BK2423 chip is the upgrade version of BK2421. The BK2423 supports not only 1Mbps and 2Mbps air rate, But also an additional 250Kbps air rate. Other functions is the same witch BK2421 . (About the similarities and differences of the BK2421 and BK2423 , please visit www.Inhaos.com for "AN0007-How to migrating code from RF-2400 to RF-2423 "file to get more deteails).

2. Hardware Structure



The demo board hardware uses ATmega324pv chip with RF-2423 wireless modules .
(Figure-01)

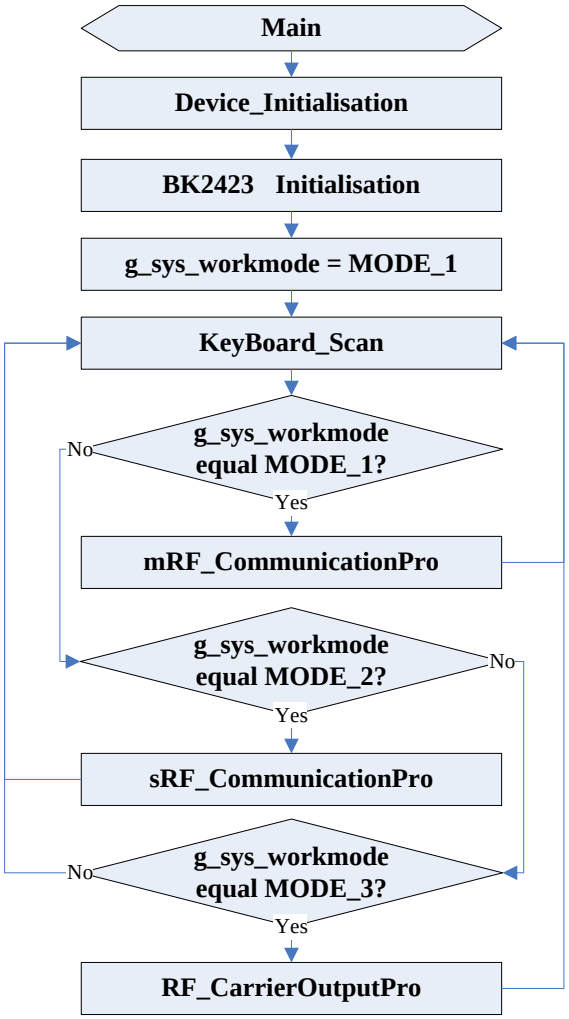
3. Software Structure

Overall software structure includes:

1. Device initialisation module: Initialize the Port, Timer and INT0 of Atmega324PV,Timer Configuration overflow period 1ms ,Initialize the Bank0 and Bank1 registers of Bk2423.

2. Key scan module: Through the key switch operating modes.Operating modes includes :Master communication mode (MODE_1) ,Slave communication mode (MODE_2) ,Carrier operating mode (MODE_3) .

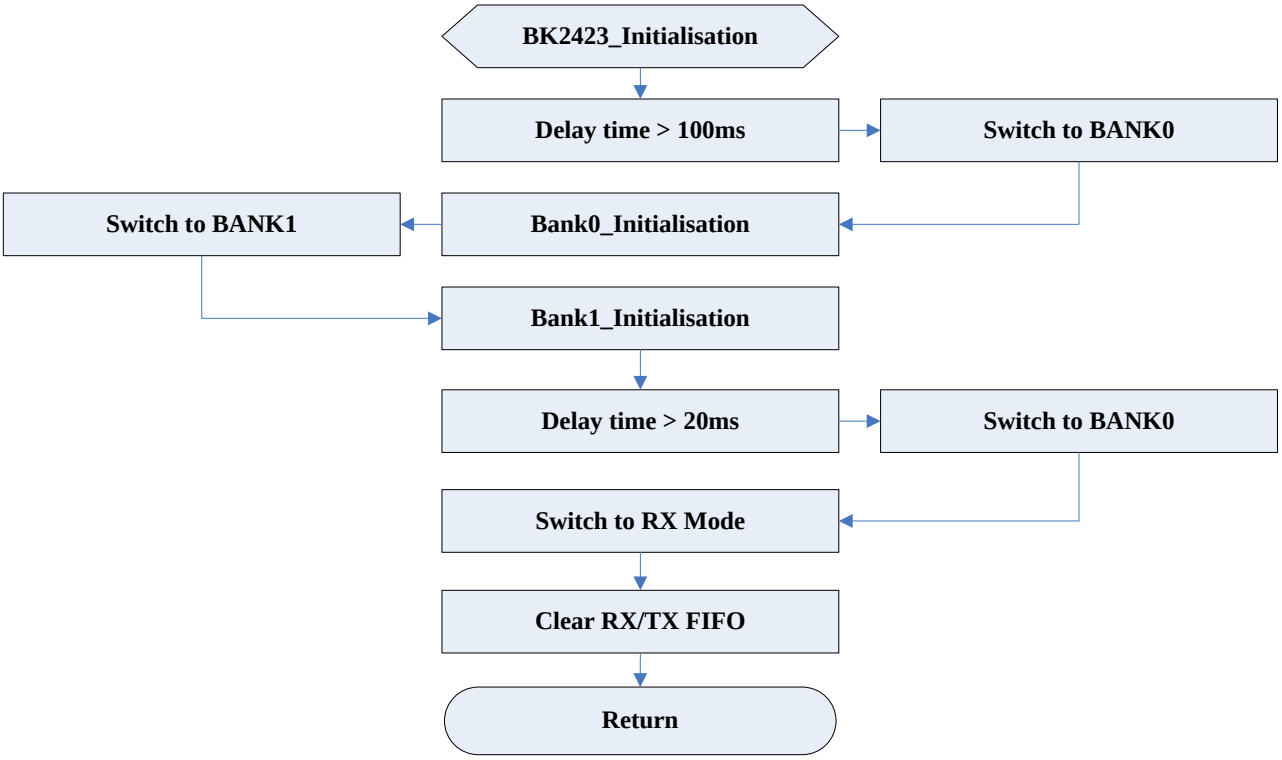
3. RF Communication module: Master Communication, Slave Communication, Carrier detection.



Main flow（Figure-02）

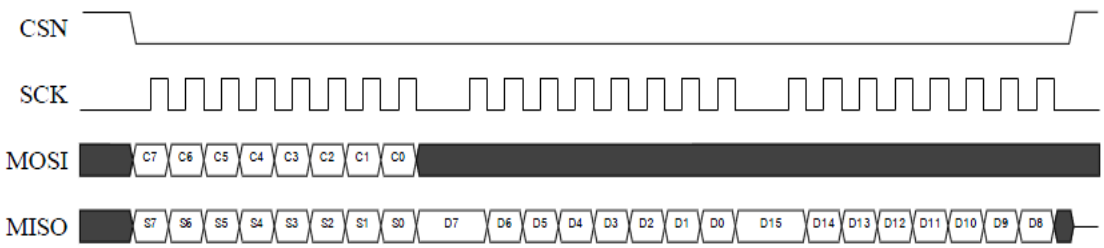
3.1 BK2423 Initialization Process

As Figure-03, BK2423 initialization process . It is the same as BK2421. All Register of BK2423 is operated through the SPI interface . Sample code reference functions "Bk2423_Init(). About more information of the BK2423 , please visit www.Inhaos.com for “BK2423 Datasheet v2.0" file to get more deteails .

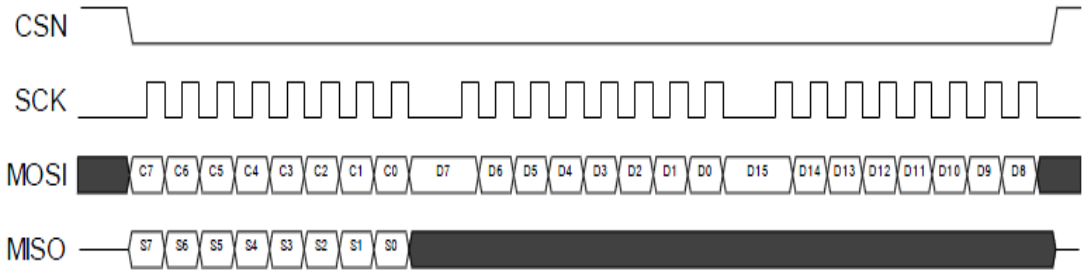


(Figure-03)

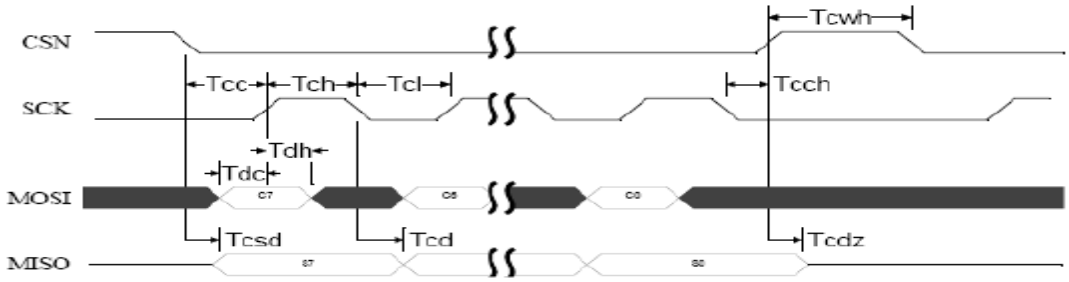
3.1.1 SPI Timing



Read Timing（Figure -04）



Write Timing（Figure -05）



（Figure-6）

Symbol	Parameters	Min	Max	Units
Tdc	Data to SCK Setup	10	/	ns
Tdh	SCK to Data Hold	2	/	ns
Tcsd	CSN to Data Valid	/	38	ns
Tcd	SCK to Data Valid	/	55	ns
Tcl	SCK Low Time	40	/	ns
Tch	SCK High Time	40	/	ns
Fsck	SCK Frequency	0	8	MHZ
Tr.Tf	SCK Rise and Fall	/	100	ns
Tcc	CSN to SCK Hold	2	/	ns
Tcch	CSN to CSN Hold	2	/	ns
Tcwh	CSN Inactive Time	50	/	ns
Tcdz	CSN to Output High Z	/	38	ns

(Form-1)

3.1.2 Initial step

Initialization Step

1. Power up
2. Delay 50 ms
3. Read current Bank, if it isn't Bank0, transfer to Bank0.
4. Write Bank0 registers, the following operation is in no order of precedence:

- CRC, interrupt mask configuration and chip power up (REG0)
- Enable Pipe (REG2)

- Channel (REG5)
- Set output power, LNA gain and air data rate (REG6)
- Set address field width (REG3)
- Enable pipe acknowledgement (REG1)
- Set pipe RX address (REG10 – REG15) and TX address (REG16)
- Set pipe payload length (REG17 – REG22)
- Set ARC and ARD (REG4), if ACK mode is enabled.
- If want to use Dynamic Payload Length or Payload With ACK, please send command ACTIVATE + 0x73 to chip. Then enable Dynamic Payload Length or Payload With ACK (REG28,REG29)

5. Transfer to Bank1.
6. Write REG0 – REG8 of Bank1 (MS byte first)
7. Write REG9 – REG13 of Bank1 (LS byte first)
8. Write REG14 of Bank1 (LS byte first)
9. Toggle REG4<25，26>, write 1 to bit25, bit 26, then write 0 to them.
10. Delay 10 ms
11. Switch to Bank0

Note : When the register configuration finished must switch to the Bank0. First you should read Bank0_Reg07_Rbank bit status value , if the RBANK bit is 0 means current state is Bank0, otherwise is Bank1 state. through the SPI write "ACTIVATE+0x53" to achieve Switch.

As Figure-07, BK2423_BANK1 register initialization value , It is the same with BK2421. “Label_1" shown that receiver sensitivity mode setting of BK2423, when set for high sensitivity mode , BK2423 receiver sensitivity will enhance 2dBm, " Label_2 " shown PLL Locking Time settings of BK2423. When set PLL Locking Time 120us , the software is compatible with the BK2421 , When set PLL Locking Time 130us , the software is compatible with nRF24L01+ . Sample code reference functions "BK2423_BANK1_Init ( ) " .

```
//In the array Bank1_Reg0_13,all[] the register value is the byte reversed!!!
volatile UINT32 Bk2423_Bank1_Reg0_13[14]={
    ////REG0 to REG8… … … … … … … … … … … … … … … … … … … … … … … … … … …
    0xE2014B40,          //REG0
    0x00004BC0,          //REG1
    0x028CFCD0,          //REG2
    0x41390099,          //REG3
    #ifdef HIGH_SENSITIVITY_DEBUG
    0x0B86BED9,          //REG4      high sensitivity mode
    #else
    0x0B869ED9,          //REG4      normal sensitivity mode
    #endif

    #ifdef  ENABLE_RSSI_DEBUG
    0xA67F023C,          //REG5      Enable RSSI measurement
    #else
    0xA67F0624,          //REG5      Disable RSSI measurement
    #endif
    0x00000000,          //REG6
    0x00000000,          //REG7
    0x00000000,          //REG8

    ////REG9 to REG13… … … … … … … … … … … … … … … … … … … … … … … … … …
    0x00000000,          //REG9
    0x00000000,          //REG10
    0x00000000,          //REG11
    #ifdef COMPATIBLE_BK2421_DEBUG
    0x00127300,          //REG12      PLL Lcoking time 120us compatible with BK2421
    #else
    0x00127305,          //REG12      PLL Lcoking time 130us compatible with nRF24L01
    #endif
    0x36B48000,          //REG13      ***********************?
};
```

Label_1

Label_2

BANK1 Initial value(Figure -07)

As Figure -05 BANK0 register initialization value , RF Initialization State in sample code as below : Transmit power for 3dBm , the data rate of 250kbps , Channel for 78 , the ordinary sensitivity mode , the address length 5 bytes, using the data channel 0 as a communication channel . Sample code reference function BK2423_BANK0_Init ( ).


```
//Bank0 register initialization value
#define BK2423_REG_NUM      23
volatile UINT8 Bk2423_Bank0_Reg[][2]={
{ 0   , 0x0F },
{ 1   , 0x00 },              //Close All Pipe ACK
{ 2   , 0x01 },              //Enalbe Pipp0 RX Address
{ 3   , 0x03 },              //RX/TX address width 5B
{ 4   , 0x00 },
{ 5   , 78   },              //Set channel
{ 6   , 0x27 },              //250K, 3DBM,High gain//0x0F or 0x2F:2Mbps; 0x07:1Mbps ; 0x27:250Kbps
{ 7   , 0x07 },
{ 8   , 0x00 },
{ 9   , 0x00 },
{ 12  , 0xc3 },
{ 13  , 0xc4 },
{ 14  , 0xc5 },
{ 15  , 0xc6 },
{ 17  , 0x20 },
{ 18  , 0x20 },
{ 19  , 0x20 },
{ 20  , 0x20 },
{ 21  , 0x20 },
{ 22  , 0x20 },
{ 23  , 0x00 },
{ 28  , 0x01 },               //Enable pipe 0, Dynamic payload length
{ 29  , 0x04 },               //EN_DPL= 1, EN_ACK_PAY = 0, EN_DYN_ACK = 0
};
```

BANK0 Initial Value(Figure -08)

3.2 Air Rate Configuration

The BK2423 supports not only 1Mbps and 2Mbps air rate, but also an additional 250Kbps air rate.Air rate can be controlled by Bank0_Reg06[3,5] = RF_DR_HIGH,RF_DR_LOW. Air rate table is as below:

AirRate	Bank0_Reg06[5] = RF_DR_LOW	Bank0_Reg06[3] = RF_DR_HIGH
2Mbps	0/1	1
1Mbps	0	0
250Kbps	1	0

(Form-2)

Air rate configuration of BK2423 is the same with BK2421. The sample code is as below

```
/**************************************************************************************
Function:                 void Set_AirRate(UINT8 Rate )
Parameter:
                          Rate      [IN]
Return:
                          None;
Description;              Bank0_Reg6[5,3]……
                          LOW,HIGH: 00          1M      0x00
                          LOW,HIGH: 01          2M      0x08
                          LOW,HIGH: 10          250K    0x20
                          LOW,HIGH: 11          2M      0x28
**************************************************************************************/
void Set_AirRate(UINT8 Rate )
```

```
{
    UINT8   Rt_Value   =   0;

    Rt_Value       =    SPI_Read_Reg( R_REGISTER | RF_SETUP );
    Rt_Value    &=   ~( (1 << 3) | (1 << 5) );

    Rt_Value    |=    Rate;
    SPI_Write_Reg( W_REGISTER | RF_SETUP,Rt_Value );

    SPI_Read_Reg(   R_REGISTER | RF_SETUP );
}
```


3.3 Output Power and Sensitivity Configuration

The receiver sensitivity of BK2423 is improved about 2dB, Meanwhile the transmitting power decreased 2dB.the default setting of Bank1_REG4 is 0x0B86BED9. Output power can be controlled byRF_PWR[2:0]. The highest bit RF_PWR[2] is Bank1_Reg4[20]; and the other 2 bits RF_PWR[1:0] are Bank0_Reg6[2:1].

Output power table is as below:

RF_PWR[2]=Bank1_REG4[20]	RF_PWR[1:0]=Bank0_REG6[2:1]	Output Power
1	11	3dBm
1	10	-2 dBm (default)
1	01	-7 dBm
1	00	-15 dBm
0	11	-25 dBm
0	10	-30 dBm
0	01	-30 dBm
0	00	-40 dBm

(Form-3)

it recommended that you should set the Bank1_REG4[20] at fisrt during initialization ,while swi tching output power simply changes the BANK0,that will improve the execution efficiency of the RF.

Switch output power mode, sample code is as below:

```
/**********************************************************************************
Function:           void BK2423_SwitchOutPower( UINT8 Power)
Parameter:
                    Power    [IN]   -15dBm; -7dBm; -2dBm; +3dBm
Return:
                    None
Description:
                    Bank1_REG4[20]   Bank0_REG6[2,1]
                        1                  00            -15dBm          0x00
                        1                  01            -7dBm           0x02
```

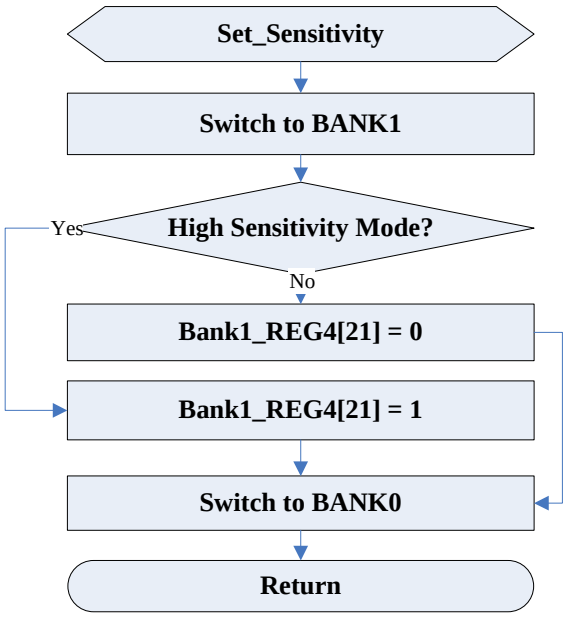
```
                                1                   10              -2dBm           0x04
                                1                   11              +3dBm           0x06
*****************************************************************************************/
void Set_OutPower( UINT8 Power)
{
    UINT8 Rt_value = 0;

    Rt_value     =   SPI_Read_Reg( R_REGISTER | RF_SETUP );

    Rt_value    &=   0xF9;
    Rt_value    |=   Power;

    SPI_Write_Reg( W_REGISTER | RF_SETUP, Rt_value );
    SPI_Read_Reg(   R_REGISTER | RF_SETUP );
}
```


Switch receive sensitivity flow chart is as Figure-06 below:



(Figure -9)

switch receive sensitivity mode,sample code is as show below:

```
/*****************************************************************************
Function:               void Set_SenMode(UINT8 b_enable )
Parameter:
                        b_enable    1:high sensitivity mode
                                    0:normal sensitivity mode
Return:

Description:
                        set sensitivity mode
*****************************************************************************/
void SPI_Bank1_Write_Reg(UINT8 reg, UINT8 *pBuf)
{
    SwitchBANK( 0x01 );         //Switch to BANK1
```

```
    SPI_Write_Buf( reg , pBuf , 0x04 );

    SwitchBANK( 0x00 );          //Switch to BANK0
}
void Set_SenMode(UINT8 b_enable )
{
    UINT8 j, WriteArr[4];

    for( j = 0x00;j < 0x04; j++ )
    {
    WriteArr[j] =    ( Bk2423_Bank1_Reg0_13[4]>>(8*(j) ) )&0xff;
    }

    if( b_enable )
    {
    WriteArr[1] =     WriteArr[1] | 0x20;              //Set REG4<21>
    }
    else
    {
    WriteArr[1] =     WriteArr[1] & 0xdf;              //Clear REG4<21>
    }

    //write REG4
    SPI_Bank1_Write_Reg( W_REGISTER|0x04 , WriteArr );
}
```


3.4 BK2423Communication Module

RF Data Transmit

1. Configuration Bank0_STATUS register PRIM_RX low, into the launch mode.
2. Before transmitting data, MCU would write the address to the TX_ADDR register, and the data to the written TX FIFO register. Note that the receiving address is the same to the receiver.
3. By raising CE, to start BK2421 to send the data in the TX FIFO. CE continued high for at least 10us.
4. After transmitting BK2421 data in Auto Answer mode (auto retransmitting count is not 0),it will immediately enter the RX mode and wait to receive ACK packets. A valid ACK packet received within the time frame means data is received successfully by receiving party. At this point, TX_DS in Bank0_STATUS register will be set to 1,while data is removed from the TX FIFO registers. If still no ACK packet in max retransmitting,BK2421 will automatically set Bank0_STATUS register MAX_RT bit to 1,and transmitting failed. The outgoing data won't be removed until software clear it.
5. CE low would enter the Standby-I mode, otherwise the system will send the next packet data in TX FIFO registers. If the register is empty and the CE is high, then enter the Standby-II model.
6. Set CE low will change Standby-II mode to Standby-I mode.

RF Data Receive

Receive mode configuration:

- Set PRIM_RX to 1 in Bank0_STATUS register
- Enable data receiving channel(by setting EA_RXADDR register)
- set the data length (set by the RX_ADDR_Pn Register)
- set RX address to the corresponding channel (through RX_ADDR_Pn register)
- set automatic answer mode (by EN_AA register)

1. Raised the CE pin to start data receiving.
2. When receiving valid data (address match, CRC checksum correctly), BK2421 stored data in RX FIFO, RX_DR bit is high, and low IRQ pin.
3. When enable the automatic answer feature,BK2421 hardware will automatically switch to transmitting mode, and launch ACK response packet (Note: the transmitting address is the same to the receiving).
4. Standby mode after CE is low.

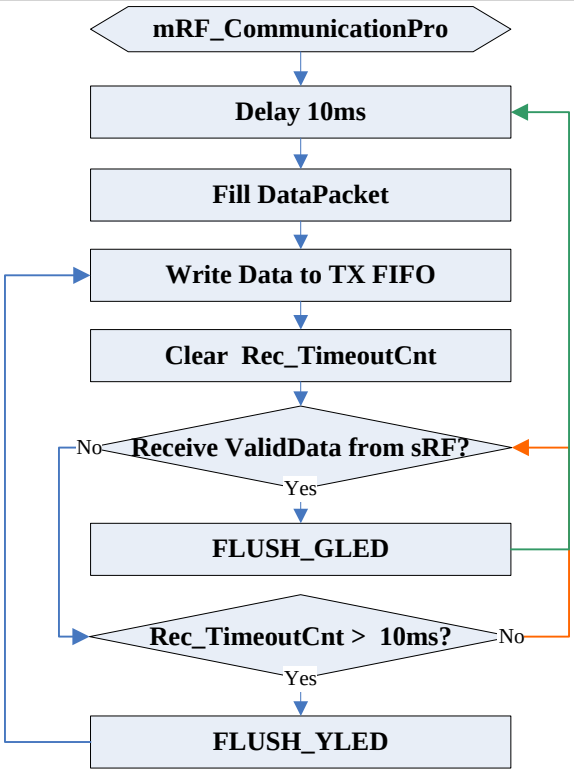
Carrier Output Module

1. Set the chip in TX mode: Write Bank0_REG [0] _BIT0 = 0, Bank0_REG0_BIT1 = 1; pull CE high.
2. Set the channel and the frequency: Write Bank0_REG [5] = 78; the corresponding frequency F = (2400 +78) MHZ.
3. Set the chip in a single carrier launch mode: Bank1_REG[04]= 0x21869ED9;
4. Set the chip in normal launch mode: Bank1_REG[04] = 0x0B869ED9
5. Sample Code reference Function “RF_CarrierOutputPro( )”

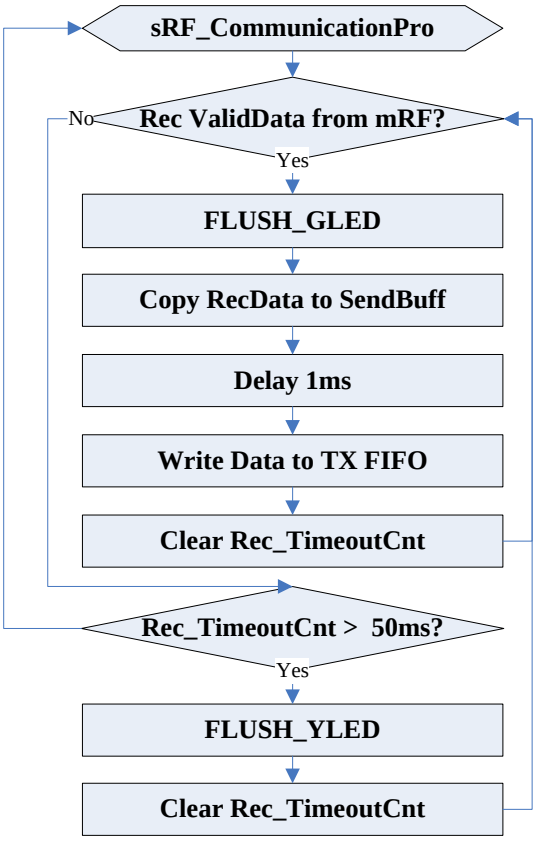
This example uses a simple one-way communication protocol .green light would flash if communication success , otherwise yellow flash.. Master communication sample code reference function mRF_Communication() , Slave communication sample code reference function “sRF_Communication ( ) . Packet format and flow chart is as below:

Bytes	ChckSum	Cmd	Sn	Length	Param[0~15]
Length	1 Byte	1 Byte	1 Byte	1 Byte	0~15 Bytes
Value	0x00~0xFF	0x00~0xFF	0x00~0xFF	0x00~0xFF	0x00~0xFF

(Form-4)



Master Flow chart（Figure -07）



Slave Flow chart（Figure -08）

4. Reference Document

1) “BK2423 Datasheet v2.0”。
2) “BK2423 Application Notes v2.0_en”。
3) “BK2421 Datasheet v2.0”。
4) “AN0007-How to migrating code from RF-2400 to RF-2423”

Declare

Due to technical limitations and the reader's understanding , this document is for reference only. Our company makes no legal commitment or guarantee of the document. If you have any doubt, please feel free to contact our company or authorized service provider, thank you! (The source code of the example can be download form www.inhaos.com.See the website for more technical support

Copyright

About Us

INHAOS is a high-tech private limited company combined with electronic products, telecommunications equipment, computer peripheral equipment development and sales. Aiming to promote domestic IT technological progress, we develop a series of embedded product development kit. This kit comes from large quantities of commercial product. The user can use it directly for design and verification, also can quickly convert the design to production and collect new product design ideas .

We also can undertake the following services:

Electronic product design

Brand components acting

Embedded development kit，Circuit module

Contact Us: http://www.inhaos.com/about.php?aID=7