



BK2401/BK2421 Programming Guide

BK2401/BK2421 编程指南

V2.0

Beken Corporation
博通集成电路(上海)有限公司
中国上海张江高科技园区
科苑路 1278 号 3A
电话: (86)21 5108 6811
传真: (86)21 6087 1277

文档含博通(BEKEN)公司保密信息, 非经书面许可, 不可外传



更改记录

版本号	日期	作者	注释
1.0	2010 年 1 月 18 日	BEKEN	正式发布
2.0	2010 年 4 月 20 日	BEKEN	修改部分错误, 更改公司新地址

CONFIDENTIAL



目录

1. 本文档主要内容.....	5
2. SPI 接口.....	5
3. 寄存器说明.....	5
3.1. BANK0 寄存器.....	5
3.2. BANK1 寄存器.....	5
4. 编程说明.....	6
4.1. 初始化.....	6
4.2. 数据包发送.....	7
4.2.1. NoACK 包.....	7
4.2.2. ACK 包.....	8
4.3. 数据包接收.....	9
5. 参考代码.....	10
5.1. 简介.....	10
5.2. 函数.....	10
5.2.1. 函数列表.....	10
5.2.2. 函数说明.....	11
5.3. 移植顺序.....	14
5.4. 其它建议.....	15



图表

图表 1 SPI 接口时序	5
图表 3 NOACK 包发送常见流程	7
图表 4 ACK 包发送常见流程	8
图表 5 数据包接收常见流程	9
图表 6 函数列表	11

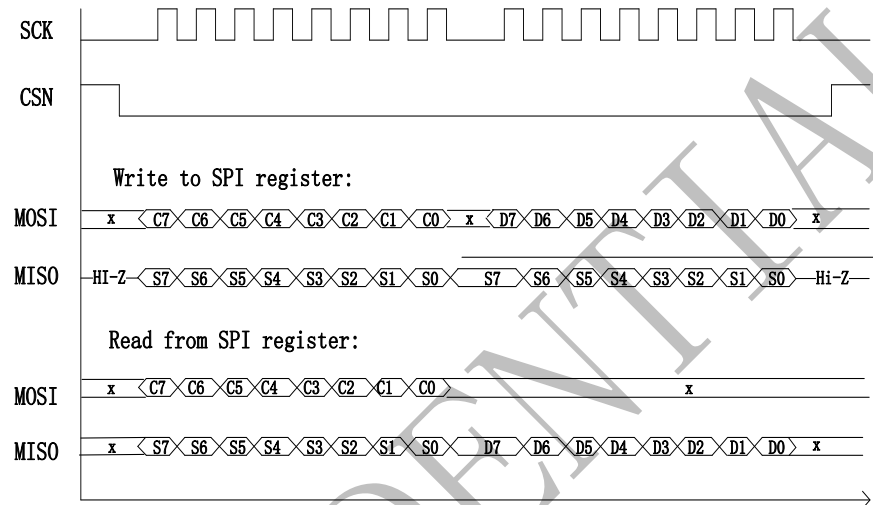
CONFIDENTIAL

1. 本文档主要内容

本文档主要目的为客户软件开发提供参考方案，其中主要包括 SPI 接口操作，编程说明和参考代码介绍。

2. SPI 接口

接口时序如下：



图表 1 SPI 接口时序

当 MCU 写 SPI 时，MCU 应该在时钟 SCK 下降沿写入数据，BK2421 会在时钟上升沿读取数据。

当 MCU 读 SPI 时，BK2421 应该在时钟 SCK 下降沿输出数据，MCU 会在时钟上升沿读取数据。

当 MCU 访问多字节的寄存器时，MCU 必须一次完成所有字节的读写。

3. 寄存器说明

BK2401/BK2421 有两组寄存器 Bank0 和 Bank1，其中 Bank 0 对应 nRF24L01 的寄存器，Bank1 是 BK2401/BK2421 的测试和功能扩展寄存器。

3.1. Bank0 寄存器

Bank0 的寄存器，详细请参见 datasheet。

3.2. Bank1 寄存器

Bank1 的寄存器，详细请参见 datasheet。

4. 编程说明

4.1. 初始化

- (1) 上电
- (2) 延时至少 50 ms
- (3) 读当前 Bank，如果当前不是 Bank0，则切换到 Bank0
- (4) 写 Bank0 寄存器，以下不分先后顺序：
 - a) CRC、中断屏蔽配置及芯片 power up (REG0)
 - b) 使能要使用的 Pipe(REG2)
 - c) 初始频点(REG5)
 - d) 设置发射功率、LNA gain、空中传输速率(REG6)
 - e) 设置数据包中的地址域长度(REG3)
 - f) 设置相应的 pipe 是否支持 acknowledgement(REG1)
 - g) 设置要使用 pipe 的 RX 地址(REG10-REG15)，及 TX 地址(REG16)
 - h) 设置要使用 pipe 的数据包长度(REG17-22)
 - i) 如果支持 ACK 模式，设置 ARC 和 ARD(REG4)
 - j) 如果要支持动态长度或者 Payload With ACK，需要先给芯片发送 ACTIVATE 命令（数据为 0x73），然后使能动态长度或者 Payload With ACK (REG28,REG29)
- (5) 切换到 Bank1
- (6) 写 Bank1 的 REG0-REG8(先写高字节，再写低字节)
- (7) 写 Bank1 的 REG9-REG13(先写低字节，再写高字节)
- (8) 写 Bank1 的 REG14(先写低字节，再写高字节)
- (9) Toggle REG4<25, 26>，即 bit25, bit26 先写 1，再写 0
- (10) 延时至少 10 ms
- (11) 切换回 Bank0

注：

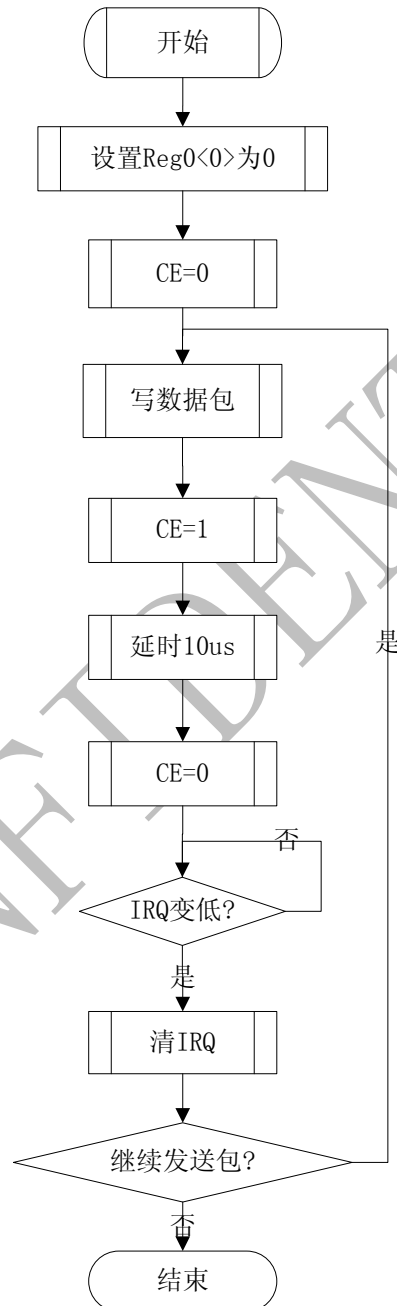
Bank1 的 REG0 到 REG8 写时序：先高字节，再低字节；每个字节从高位再到低位。

但是 Bank1 的 REG9 到 REG14 及 Bank0 的读写时序和 nRF24L01 一致，都是先低字节，再高字节；每个字节仍然从高位到低位。

4.2. 数据包发送

4.2.1. NoACK 包

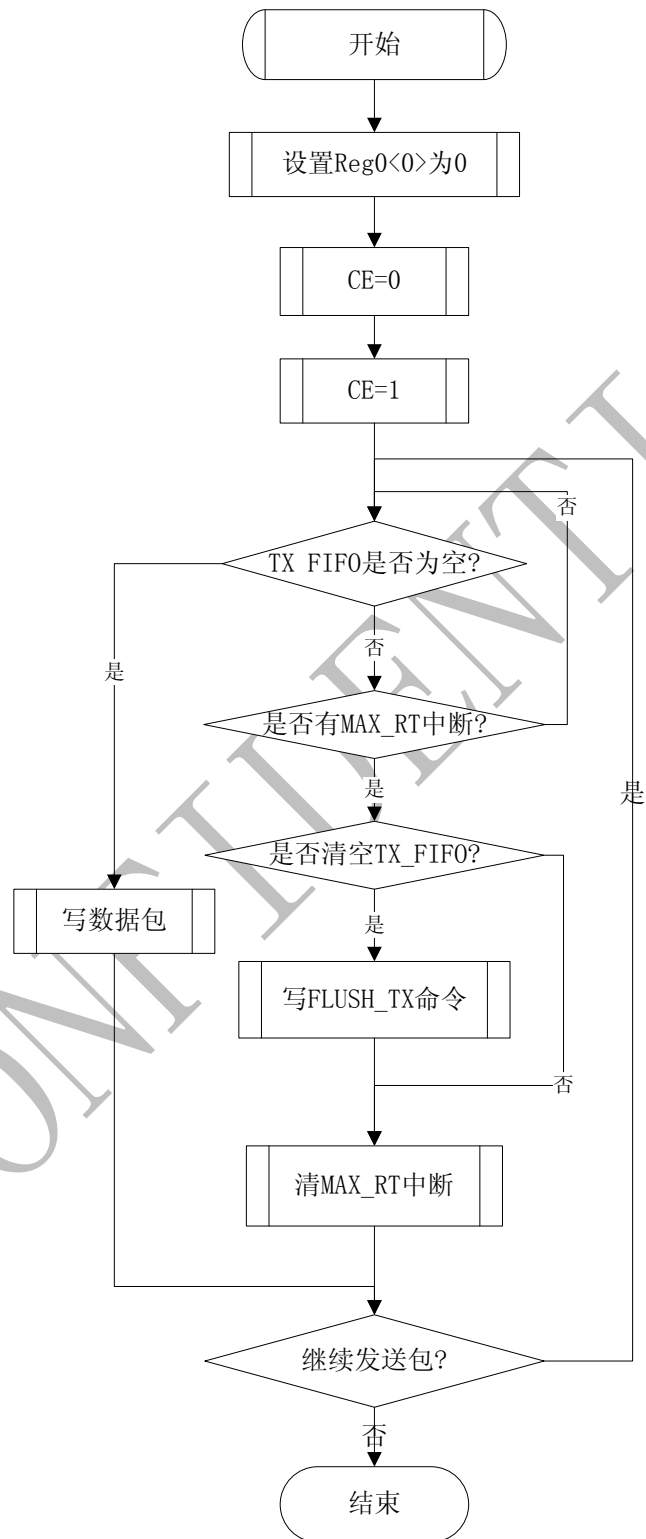
图表 2 为 NoACK 包发送的一般流程。



图表 2 NoACK 包发送常见流程

4.2.2. ACK 包

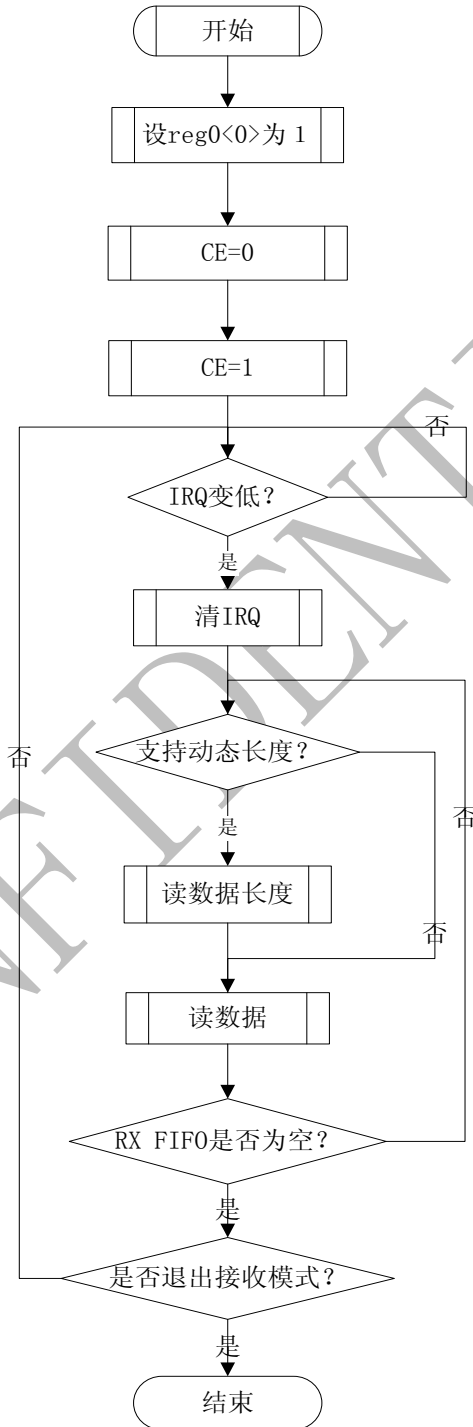
图表 3 为 ACK 包发送的常见流程。



图表 3 ACK 包发送常见流程

4.3. 数据包接收

图表 4 为数据包的常见接收流程。



图表 4 数据包接收常见流程

5. 参考代码

5.1. 简介

(1) 参考代码包含以下 4 个文件：

a) BK2421_Initialize.c:

提供寄存器读写，芯片初始化，收发模式切换和 channel 切换等函数。

b) BK2421_TxRx.c:

提供收包和发包的函数。

c) Test_Func.c:

提供载波发射，BER 测试等测试和扩展函数。

d) BK2421.h:

包含宏定义等的头文件

(2) 参考代码在 Keil C51 uVision2 环境下编译通过。

(3) 参考代码提供的初始值是常用的典型设置(支持 6 个 pipe，动态长度，NoACK 和 ACK 模式，Payload With ACK 模式，2 bytes 的 CRC，1M 模式，0dBm 输出功率)。

(4) 用户应根据实际需要对代码进行移植并做相应修改和功能添加。

注：BK2401/BK2421 的软件编程中，必须要考虑躲避干扰问题，请参考 *BK2401_BK2421 Application Notes* 相关章节。

5.2. 函数

5.2.1. 函数列表

SPI_RW	写一个 byte 到芯片，并返回一个 byte
SPI_WRITE_Reg	写一个 byte 到一个寄存器
SPI_Read_Reg	从寄存器读一个 byte
SPI_Read_Buf	从寄存器读多个 byte
SPI_Write_Buf	写多个 byte 到一个寄存器
SwitchToRxMode	切换芯片到 RX 模式
SwitchToTxMode	切换芯片到 TX 模式
BK2421_Initialize	芯片初始化，并进入到 RX 模式
Send_Packet	发送包
Receive_Packet	接受包

SetChannelNum	设置新频点
SwitchCFG	切换到 Bank1 寄存器操作，仅为载波测试、BER 测试或者其它特殊功能时使用
Carrier_Test	载波测试 开始发射载波时，调用 Carrier_Test(1) 停止发射载波时，调用 Carrier_Test(0)
BER_Test	调用该函数可以读出硬件计算出的 BER 数值 输入参数 ms: 为硬件 BER 持续的毫秒数 输出参数 received_total_bits: 为在 ms 时间内，硬件接收到的所有 bits 数目 输出参数 received_error_bits: 为在 ms 时间内，硬件接收到的错误 bits 数目 received_error_bits/received_total_bits 即为 BER
SPI_Analog_Write_Reg	写 Bank1 寄存器
SPI_Analog_Read_Reg	读 Bank1 寄存器
Set_LowPower_Mode	芯片进入低输出 power 模式
Close_CD_Detect	如果不需要 CD 功能，可以关闭该功能，减少大约 1mA 电流
Get_Chip_ID	得到芯片的 ID 号

图表 5 函数列表

注:背景带蓝色的部分涉及到读写 Bank1 的寄存器，建议客户根据需要直接使用或者移植相关代码。

5.2.2. 函数说明

(1) **UINT8 SPI_RW(UINT8 value)**

描述: 通过 SPI 的 MOSI 线向芯片写入 8 个 bit 数，并从 MISO 线上获取 8 个 bit。

参数: value: 向 MOSI 线上输出的 8bit 数

返回值: 从 MISO 线上获取的 8bit 数

(2) **void SPI_Write_Reg(UINT8 reg, UINT8 value)**

描述: 通过 SPI 向芯片的某个寄存器写入一个字节。

参数: reg: 需要写入的寄存器序号

value: 需要写入的值

返回值: 无

(3) UINT8 SPI_Read_Reg(UINT8 reg)

描述：通过 SPI 从芯片的某个寄存器读出一个字节。

参数：reg: 需要读取的寄存器序号

返回值：读出的寄存器值

(4) void SPI_Read_Buf(UINT8 reg, UINT8 *pBuf, UINT8 length)

描述：通过 SPI 从多字节的寄存器中读出多个字节数据。

参数：reg: 需要读取的寄存器序号

 pBuf: 数据存放的数组

 length: 需要读取的数据长度

返回值：无

(5) void SPI_Write_Buf(UINT8 reg, UINT8 *pBuf, UINT8 length)

描述：通过 SPI 的 MOSI 线向芯片写入 8 个 bit 数，并从 MISO 线上获取 8 个 bit。

参数：reg: 需要写入的寄存器序号

 pBuf: 数据存放的数组

 length: 需要写入的数据长度

返回值：无

(6) void SwitchToRxMode()

描述：将芯片切换为接收模式。

参数：无

返回值：无

(7) void SwitchToTxMode()

描述：将芯片切换为发送模式。

输入：无

返回值：无

(8) void BK2421_Initialize()

描述：完成芯片初始化，配置所有寄存器，并切换为接收模式。

参数：无

返回值：无

(9) void Send_Packet(UINT8 type,UINT8* pbuf,UINT8 len)

描述：发送一个数据包。

参数：type: WR_TX_PLOAD: 以带 ACK 模式发送数据包
 W_TX_PAYLOAD_NOACK_CMD: 以 NOACK 模式发送数据包
 pbuf: 数据所存放的数组
 len: 数据包的长度

返回值：无

(10) void Receive_Packet()

描述：循环检测状态寄存器，若有接收到数据，则将数据从 FIFO 中读出。

参数：无

返回值：无

(11) void SetChannelNum(UINT8 ch)

描述：设置 channel 号。

参数：ch: 0-127: channel 号

返回值：无

(12) void SwitchCFG(char _cfg)

描述：寄存器 bank 切换。

参数：_cfg: 0: 切换到 bank0; 1: 切换到 bank1

返回值：无

(13) void Carrier_Test(UINT8 b_enable)

描述：发射连续载波模式。

参数：b_enable: 0: 打开连续载波模式; 1: 关闭连续载波模式。

返回值：无

(14) void BER_Test(UINT16 ms,UINT32* received_total_bits,UINT32* received_error_bits)

描述：BER 测试函数。

参数：ms: BER 测试进行的时间，单位为毫秒。
 received_total_bits: 存放测试时间内接收到的总 bit 数的指针。
 received_error_bits: 存放测试时间内接收到的错误 bit 数的指针。

返回值：无

(15) void SPI_Bank1_Write_Reg(UINT8 reg, UINT8 *pBuf)

描述：向 bank1 寄存器写入数据。

参数：reg: 需要写入的寄存器序号

pBuf: 数据存放的数组

返回值：无

(16) void SPI_Bank1_Read_Reg(UINT8 reg, UINT8 *pBuf)

描述：从 bank1 寄存器中读出数据。

参数：reg: 需要读取的寄存器序号

pBuf: 数据存放的数组

返回值：无

(17) void Set_LowPower_Mode()

描述：进入低功率模式。

参数：无

返回值：无

(18) void Close_CD_Detect()

描述：关闭 CD 检测功能。

参数：无

返回值：无

(19) UINT8 Get_Chip_ID()

描述：读取芯片的 ID。

参数：无

返回值：芯片的 ID 号

5.3. 移植顺序

- (1) 添加 SPI_RW, SPI_WRITE_Reg, SPI_Read_Reg, SPI_Write_Buf, SPI_Read_Buf 函数，调试 SPI 读写。例如，写 REG0，然后读出来，看看是否是写入的值
- (2) 添加 Initialize 相关的代码
- (3) 调用 Send_Packet 和 Receive_Packet，看看能否收发包
- (4) 根据应用需要，增加其它功能

5.4. 其它建议

- (1) 建议调试程序时，最好是先和我们的评估板互通。客户程序先发送，评估板接收；然后用评估板发送，客户程序接收。
- (2) 寄存器的初始值最好先用我们提供的初始值，该值和我们评估板的初始值一样。程序能够收发包后，再修改初始值为产品需要的。

CONFIDENTIAL