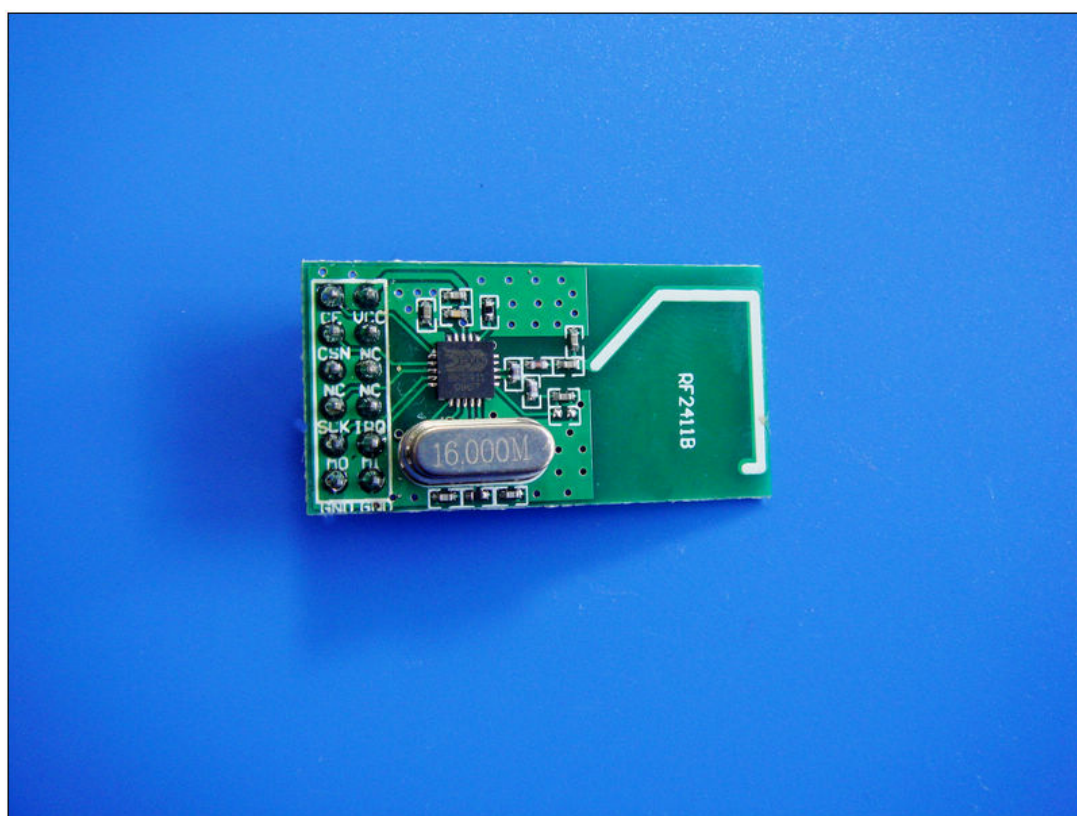


BK2411 无线模块

用户手册



模块尺寸 (PCB 板载天线 1mm*17.5mm*34mm)

目录

产品概述	3
基本特点	3
引脚接口说明	4
BK2411 工作方式	5
TX模式	6
RX模式	7
空闲模式	8
关机模式	8
BK2411 模块参数配置	9
SPI指令	9
编程指南	17
程序分析	17
命令寄存器宏定义	17
工作方式配置寄存器宏定义	18
状态寄存器宏定义	19
自动应答寄存器宏定义	19
BK2411 SPI读写时序操作	19
BK2411 SPI写寄存器操作	20
BK2411 SPI读寄存器操作	20
工作模式初始化	20
BK2411 发送模式设置	21
BK2411 数据发送流程	21
BK2411 接收模式设置	22
BK2411 接收数据流程	22
无线应用注意事项	22
我们的承诺	24

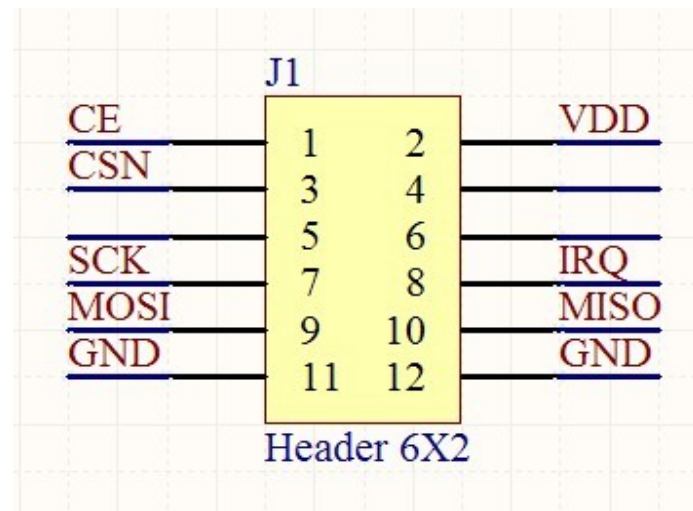
产品概述

BK2411 是由博通（BEKEN）推出的一款国产 2.4G 无线收发芯片，和 nRF24L01 硬件引脚兼容，软件上和 nRF24L01 也是基本相同，可以说是 nRF24L01 的国产仿制品，针对工业控制、门禁、考勤、监控、安防等行业应用特点推出。

基本特点

- (1) 2.4Ghz 全球开放 ISM 频段免许可证使用
- (2) FSK 调制使其灵敏度更好，GFSK 调制使其频率更有效率
- (3) 拥有 1Mbps 或者 2Mbps 的空中数据传送速率
- (4) 可编程的输出功率有：-35，-25，-15，-5，0，5 dBm
- (5) 在 0 dBm 的输出功率下，发送模式消耗 14mA 的功耗；
- (6) 在 1Mbps 数据传送速率下，接受模式消耗 21mA 的功耗
- (7) 允许+/- 60ppm 16 MHz 的晶振
- (8) 数据包长度可以 1 到 32 字节。
- (9) 自动数据处理功能
- (10) 适合于 1：6 星形网络的 6 数据通道
- (11) 供应电源为 1.9V 到 3.6V
- (12) 在 POWERDOWN 模式下为 3uA 的直流电流，在 standby-I 模式下为 410uA 直流电流
- (13) 带有最大时钟速率为 8Mbps MCU 的 4 针 SPI 接口

引脚接口说明



引脚功能

管脚	名称	管脚功能	备注说明
1	CE	数字输入	收发模式切换控制端口
2	VDD	电源	1.9-3.6 V
3	CSN	数字输入	SPI片选使能端，低电平有效
4-6	NC	无	悬空
7	SCK	数字输入	SPI时钟
8	IRQ	数字输出	工作状态指示
9	MOSI	数字输入	SPI从设备数据输入
10	MISO	数字输出	SPI从设备数据输出
11	GND	电源地	和系统共地
12	GND	电源地	和系统共地

备注

1. VCC 引脚的电压范围为2.3-3.6V 之间，不能在这个区间之外，如超过 3.6V 将会烧毁模块。推荐电压 3.3V 左右；
2. 硬件没有集成SPI功能的单片机也可以控制本模块，用普通单片I/O口模拟 SPI 时序进行读写操作即可；
3. 模块接口采用标准2.54mmDIP插针，13 脚、14 脚为接地脚，需要和系统电路的逻辑地连接起来；
4. 与 51 系列单片机 P0 口连接时候，需要加 10K 的上拉电阻，与其余口连接不需要。其他系列的5V单片机，如AVR、PIC，请参考该系列单片机 I/O 口输出电流大小，如果超过 10mA，需要串联2-5K电阻分压，否则容易烧毁模块！如果是 3.3V 的MCU，可以直接和模块的I/O口线连接。

BK2411 工作方式

BK2411 有工作模式有四种：

收发模式

配置模式

空闲模式

关机模式

而这些工作模式由 PWR_UP register 、 PRIM_RX register 和 CE 决定，详见下表。

Mode	PWR_UP register	PRIM_RX register	CE	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFO
TX mode	1	0	1→0	Stays in TX mode until packet transmission is finished
Standby-II	1	0	1	TX FIFO empty
Standby-I	1	-	0	No ongoing packet transmission
Power Down	0	-	-	-

收发模式

TX 模式

PTX 设备（PRIM_RX=0）：

当 PTX 设备发送数据包时，发送模式被激活。为了在 POWER DOWN 模式进入 TX 模式，PTX 设备必须使 PWR_UP 位置高，PRIM_RX 位置低，在 TX FIFO 中有有效数据，在 CE 引脚有超过 10us 的高电平脉冲。

PTX 设备停留在 TX 模式直到已经完成当前数据包的传送。如果 CE=0 它将返回到 standby-I 模式。如果 CE=1, 则下一个行为由 TX FIFO 中的状态决定。如果 TX FIFO 不是空的，则 PTX 设备停留在 TX 模式，并传送下一个数据包。如果 TX FIFO 是空的，则 PTX 设备将进入 standby-II 模式。

如果自动重新传输使能（EN_AA=1）和自动应答被要求（NO_ACK=0），则当 ARD 通过后和重新发送计数小于 ARC 时，PTX 设备将从 standby-I 模式进入 TX 模式。

PTX 设备（PRIM_RX=1）：

只有当 EN_AA=1 和 NO_ACK=0 在接收数据时发送确认在 TX FIFO 中有有效数据的数据包,PTX 设备将有 RX 模式进入 TX 模式。

RX 模式

PRX 设备 (PRIM_RX=1) :

当 BK2411 射频被配置为接收器时, RX 模式将被激活。为了从 standby-I 模式进入 RX 模式, PRX 设备必须使 PWR_UP 位置高, PRIM_RX 位置高, CE 引脚置高。或者当在接收数据包时 EN_AA=1 和 NO_ACK=0 的情况下, PRX 设备能够从在 TX 模式发送完确认数据包后进入 RX 模式。

在 RX 模式下接收机从 RF 通道解调信号并且不断地发出解调数据给数据包处理引擎。数据包处理引擎不断寻找有效数据包。如果有效数据包被找到 (通过匹配地址和有效 CRC 校验), 则有效数据包将会出现在 RX FIFO 的空闲位置处。如果 RX FIFO 满了, 则接收数据包被抛弃。

PRX设备停留在RX模式直到MCU配置为standby-I模式或power down模式

在 RX 模式下载波检测 (CD) 可用。当 RF 信号在接收频率通道被检测到时, CD 位将被置高。为了安全检测, 信号必须用 FSK/GFSK 进行调制。同时其他的信号也会被检测到。当 RF 信号在 RX 模式下被检测到时 CD 位置高, 否则 CD 将会置低。在 CD 寄存器可用之前, 内部 CD 信号将被过滤掉。在 CD 置高后大约 200us

后 RF 信号必须发出。

PTX 设备 (PRIM_RX=0) :

只有当 EN_AA=1 和 NO_ACK=0 的情况下接收确认数据包时 PTX 设备才会从 TX 模式进入 RX 模式。

空闲模式

BK2411 的空闲模式是为了减小平均工作电流而设计，其最大的优点是，实现节能的同时，缩短芯片的起动时间。在空闲模式下，部分片内晶振仍在工作，此时的工作电流跟外部晶振的频率有关。

关机模式

在关机模式下，为了得到最小的工作电流，一般此时的工作电流为 900nA 左右。关机模式下，其配置字的内容也会被保持在 BK2411 片内，这是该模式与断电状态最大的区别。

BK2411 模块参数配置

BK2411 的所有配置工作都是通过 SPI 完成。工作于数据包收发模式，这种工作模式下，系统的程序编制会简单，且稳定性也会更高，因此，下文着重介绍如何把 BK2411 配置为数据包收发模式，从而掌握无线数据控制操作方法。数据包的配置字使 BK2411 能够自动处理射频协议，在配置完成后，在 BK2411 工作的过程中，只需改变其最低一个字节中的内容，以实现接收模式和发送模式之间切换。

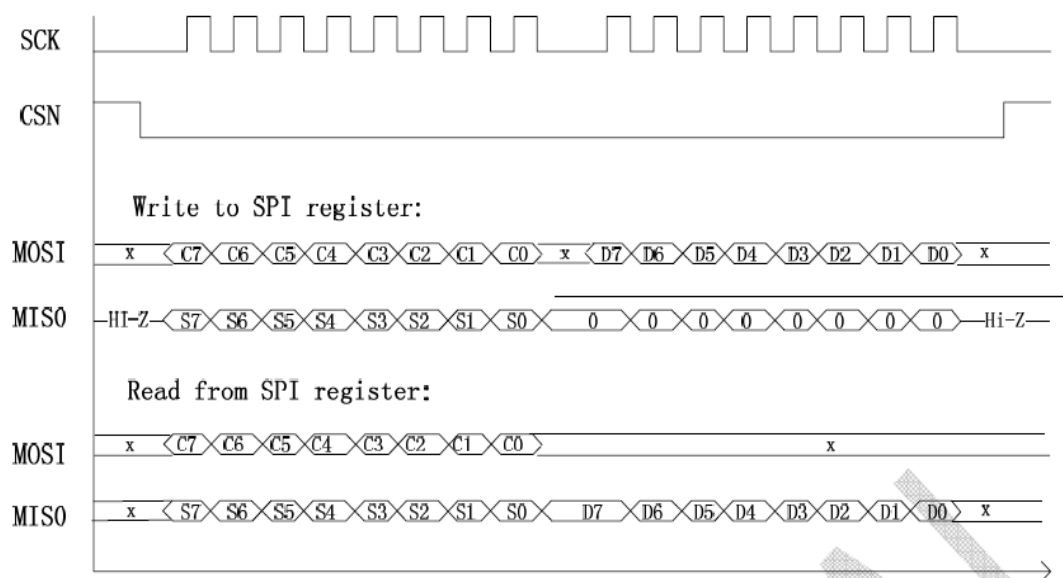
SPI 指令

每个新命令必须通过 CSN 的下降沿来使能。并行的 SPI 命令字通过 MOSI 引脚应用，在 MISO 引脚状态寄存器通过移位逐次地输出，SPI 指令如下表：

Command name	Command word (binary)	# Data bytes	Operation
R_REGISTER	000A AAAA	1 to 5 LSB byte first	Read command and status registers. AAAA = 5 bit Register Map Address
W_REGISTER	001A AAAA	1 to 5 LSB byte first	Write command and status registers. AAAA = 5 bit Register Map Address Executable in power down or standby modes only.
R_RX_PAYLOAD	0110 0001	1 to 32 or 1-255 LSB byte first	Read RX-payload: 1 – 32 bytes or 1-255 bytes. A read operation always starts at byte 0. Payload is deleted from FIFO after it is read. Used in RX mode.
W_TX_PAYLOAD	1010 0000	1 to 32 or 1 to 255 LSB byte first	Write TX-payload: 1 – 32 bytes or 1-255 bytes. A write operation always starts at byte 0 used in TX payload.
FLUSH_TX	1110 0001	0	Flush TX FIFO, used in TX mode
FLUSH_RX	1110 0010	0	Flush RX FIFO, used in RX mode Should not be executed during transmission of acknowledge, that is, acknowledge package will not be completed.
REUSE_TX_PL	1110 0011	0	Used for a PTX device Reuse last transmitted payload. Packets are repeatedly retransmitted as long as CE is high. TX payload reuse is active until W_TX_PAYLOAD or FLUSH TX is executed. TX payload reuse must not be activated or deactivated during package transmission

ACTIVATE	0101 0000	1	This write command followed by data 0x73 activates the following features: • R_RX_PL_WID • W_ACK_PAYLOAD • W_TX_PAYLOAD_NOACK A new ACTIVATE command with the same data deactivates them again. This is executable in power down or stand by modes only. The R_RX_PL_WID, W_ACK_PAYLOAD, and W_TX_PAYLOAD_NOACK features registers are initially in a deactivated state; a write has no effect, a read only results in zeros on MISO. To activate these registers, use the ACTIVATE command followed by data 0x73. Then they can be accessed as any other register in BK2411. Use the same command and data to deactivate the registers again. This write command followed by data 0x53 toggles the register bank, and the current register bank number can be read out from REG7 [7]
R_RX_PL_WID	0110 0000		Read RX-payload width for the top R_RX_PAYLOAD in the RX FIFO.
W_ACK_PAYLOAD	1010 1PPP	1 to 32 or 1 to 255 LSB byte first	Used in RX mode. Write Payload to be transmitted together with ACK packet on PIPE PPP. (PPP valid in the range from 000 to 101). Maximum three ACK packet payloads can be pending. Payloads with same PPP are handled using first in - first out principle. Write payload: 1–32 bytes or 1–255 bytes. A write operation always starts at byte 0.
W_TX_PAYLOAD_NOACK	1011 000	1 to 32 or 1 to 255 LSB byte first	Used in TX mode. Disables AUTOACK on this specific packet.
NOP	1111 1111	0	No Operation. Might be used to read the STATUS register

SPI读写时序图



BK2411 配置字的 CONFIG 寄存器的位描述如下表所示。

Register bank 0:

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
00	CONFIG				Configuration Register
	Reserved	7	0	R/W	Only '0' allowed
	MASK_RX_DR	6	0	R/W	Mask interrupt caused by RX_DR 1: Interrupt not reflected on the IRQ pin 0: Reflect RX_DR as active low interrupt on the IRQ pin
	MASK_TX_DS	5	0	R/W	Mask interrupt caused by TX_DS 1: Interrupt not reflected on the IRQ pin 0: Reflect TX_DS as active low interrupt on the IRQ pin
	MASK_MAX_RT	4	0	R/W	Mask interrupt caused by MAX_RT 1: Interrupt not reflected on the IRQ pin 0: Reflect MAX_RT as active low interrupt on the IRQ pin
	EN_CRC	3	1	R/W	Enable CRC. Forced high if one of the bits in the EN_AA is high
	CRCO	2	1	R/W	CRC encoding scheme '0' - 1 byte '1' - 2 bytes
	PWR_UP	1	1	R/W	1: POWER UP, 0: POWER DOWN
	PRIM_RX	0	0	R/W	RX/TX control, only can be switched in power down state (REG0[1] = 0) 1: PRX, 0: PTX
01	EN_AA				Enable 'Auto Acknowledgment' Function
	Reserved	7:6	00	R/W	Only '00' allowed
	ENAA_P5	5	1	R/W	Enable auto acknowledgement data pipe 5
	ENAA_P4	4	1	R/W	Enable auto acknowledgement data pipe 4
	ENAA_P3	3	1	R/W	Enable auto acknowledgement data pipe 3
	ENAA_P2	2	1	R/W	Enable auto acknowledgement data pipe 2
	ENAA_P1	1	1	R/W	Enable auto acknowledgement data pipe 1
	ENAA_P0	0	1	R/W	Enable auto acknowledgement data pipe 0
02	EN_RXADDR				Enabled RX Addresses
	Reserved	7:6	00	R/W	Only '00' allowed
	ERX_P5	5	0	R/W	Enable data pipe 5.
	ERX_P4	4	0	R/W	Enable data pipe 4.
	ERX_P3	3	0	R/W	Enable data pipe 3.
	ERX_P2	2	0	R/W	Enable data pipe 2.
	ERX_P1	1	1	R/W	Enable data pipe 1.
	ERX_P0	0	1	R/W	Enable data pipe 0.
03	SETUP_AW				Setup of Address Widths (common for all data pipes)

	Reserved	7:2	000000	R/W	Only '000000' allowed
	AW	1:0	11	R/W	RX/TX Address field width '00' - Illegal '01' - 3 bytes '10' - 4 bytes '11' - 5 bytes LSB byte is used if address width is below 5 bytes
04	SETUP_RETR				Setup of Automatic Retransmission
	ARD	7:4	0000	R/W	Auto Retransmit Delay '0000' - Wait 250 uS '0001' - Wait 500 uS '0010' - Wait 750 uS '1111' - Wait 4000 us (Delay defined from end of transmission to start of next transmission)a
	ARC	3:0	0011	R/W	Auto Retransmit Count '0000' - Re-Transmit disabled '0001' - Up to 1 Re-Transmit on fail of AA '1111' - Up to 15 Re-Transmit on fail of AA
05	RF_CH				RF Channel
	Reserved	7	0	R/W	Only '0' allowed
	RF_CH	6:0	0000010	R/W	Sets the frequency channel BK2411 operates on
06	RF_SETUP				RF Setup Register
	Reserved	7	111	R/W	Only '000' allowed
	RSSI_EN	6	1	R/W	Enable RSSI measurement 0: Disable 1: Enable
	DREG_ON	5	1	R/W	Digital regulator can be shut down or not 0: Can be shut down in stand-by I mode 1: Always on in any state except power down
	RF_PWR[2]	4	1	R/W	RF output power in TX mode
	RF_DR	3	1	R/W	Air Data Rate '0' - 1Mbps '1' - 2Mbps Register 4 and 13 in bank 1 have different setting in 1 Mbps and 2 Mbps data rate mode
	RF_PWR[1:0]	2:1	11	R/W	Set RF output power in TX mode RF_PWR[2:0] '000' - -35dBm '001' - -25 dBm '010' - -15 dBm '011' - -5 dBm '100' - -5 dBm '101' - -5 dBm '110' - 0 dBm '111' - 5 dBm
	LNA_HCURR	0	1	R/W	Setup LNA gain

07	STATUS				Status Register (In parallel to the SPI command word applied on the MOSI pin, the STATUS register is shifted serially out on the MISO pin)
	RBANK	7	0	R	Register bank selection states. Switch register bank is done by SPI command "ACTIVATE" followed by 0x53 0: Register bank 0 1: Register bank 1
	RX_DR	6	0	R/W	Data Ready RX FIFO interrupt Asserted when new data arrives RX FIFO Write 1 to clear bit.
	TX_DS	5	0	R/W	Data Sent TX FIFO interrupt Asserted when packet transmitted on TX. If AUTO_ACK is activated, this bit is set high only when ACK is received. Write 1 to clear bit.
	MAX_RT	4	0	R/W	Maximum number of TX retransmits interrupt Write 1 to clear bit. If MAX_RT is asserted it must be cleared to enable further communication.
	RX_P_NO	3:1	111	R	Data pipe number for the payload available for reading from RX_FIFO 000-101: Data Pipe Number 110: RX_FIFO Empty 111: Not used
	TX_FULL	0	0	R	TX_FIFO full flag. 1: TX_FIFO full 0: Available locations in TX_FIFO
08	OBSERVE_TX				Transmit observe register
	PLOS_CNT	7:4	0	R	Count lost packets. The counter is overflow protected to 15, and discontinues at max until reset. The counter is reset by writing to RF_CH.
	ARC_CNT	3:0	0	R	Count retransmitted packets. The counter is reset when transmission of a new packet starts.
09	CD				
	Reserved	7:1	000000	R	
	CD	0	0	R	Carrier Detect
0A	RX_ADDR_P0	39:0	0xE7E7E7E7	R/W	Receive address data pipe 0. 5 Bytes maximum length. (LSB byte is written first. Write the number of bytes defined by SETUP_AW)
0B	RX_ADDR_P1	39:0	0xC2C2C2C2	R/W	Receive address data pipe 1. 5 Bytes maximum length. (LSB byte is written first. Write the number of bytes defined by SETUP_AW)
0C	RX_ADDR_P2	7:0	0xC3	R/W	Receive address data pipe 2. Only LSB MSB bytes is equal to RX_ADDR_P1[39:8]
0D	RX_ADDR_P3	7:0	0xC4	R/W	Receive address data pipe 3. Only LSB MSB bytes is equal to RX_ADDR_P1[39:8]
0E	RX_ADDR_P4	7:0	0xC5	R/W	Receive address data pipe 4. Only LSB. MSB bytes is equal to RX_ADDR_P1[39:8]
0F	RX_ADDR_P5	7:0	0xC6	R/W	Receive address data pipe 5. Only LSB.

					MSB bytes is equal to RX_ADDR_P1[39:8]
10	TX_ADDR	39:0	0xE7E7E7E7	R/W	Transmit address. Used for a PTX device only. (LSB byte is written first) Set RX_ADDR_P0 equal to this address to handle automatic acknowledge if this is a PTX device
11	RX_PW_P0				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P0	5:0	0	R/W	Number of bytes in RX payload in data pipe 0 (1 to 32 bytes). 0: not used 1 = 1 byte ... 32 = 32 bytes
12	RX_PW_P1				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P1	5:0	0	R/W	Number of bytes in RX payload in data pipe 1 (1 to 32 bytes). 0: not used 1 = 1 byte ... 32 = 32 bytes
13	RX_PW_P2				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P2	5:0	0	R/W	Number of bytes in RX payload in data pipe 2 (1 to 32 bytes). 0: not used 1 = 1 byte ... 32 = 32 bytes
14	RX_PW_P3				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P3	5:0	0	R/W	Number of bytes in RX payload in data pipe 3 (1 to 32 bytes). 0: not used 1 = 1 byte ... 32 = 32 bytes
15	RX_PW_P4				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P4	5:0	0	R/W	Number of bytes in RX payload in data pipe 4 (1 to 32 bytes). 0: not used 1 = 1 byte ... 32 = 32 bytes
16	RX_PW_P5				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P5	5:0	0	R/W	Number of bytes in RX payload in data pipe 5 (1 to 32 bytes). 0: not used 1 = 1 byte

					...
					32 ■ 32 bytes
17	FIFO_STATUS				FIFO Status Register
	Reserved	7	0	R/W	Only '0' allowed
	TX_REUSE	6	0	R	Reuse last transmitted data packet if set high. The packet is repeatedly retransmitted as long as CE is high. TX_REUSE is set by the SPI command REUSE_TX_PL, and is reset by the SPI commands W_TX_PAYLOAD or FLUSH TX
	TX_FULL	5	0	R	TX FIFO full flag 1: TX FIFO full; 0: Available locations in TX FIFO
	TX_EMPTY	4	1	R	TX FIFO empty flag. 1: TX FIFO empty 0: Data in TX FIFO
	Reserved	3:2	00	R/W	Only '00' allowed
	RX_FULL	1	0	R	RX FIFO full flag 1: RX FIFO full 0: Available locations in RX FIFO
	RX_EMPTY	0	1	R	RX FIFO empty flag 1: RX FIFO empty 0: Data in RX FIFO
N/A	ACK_PLD	255:0	X	W	Written by separate SPI command ACK packet payload to data pipe number PPP given in SPI command Used in RX mode only Maximum three ACK packet payloads can be pending. Payloads with same PPP are handled first in first out.
N/A	TX_PLD	255:0	X	W	Written by separate SPI command TX data payload register 1 - 32 bytes. This register is implemented as a FIFO with three levels. Used in TX mode only
N/A	RX_PLD	255:0	X	R	Read by separate SPI command RX data payload register. 1 - 32 bytes. This register is implemented as a FIFO with three levels. All RX channels share the same FIFO.
IC	DYNPD				Enable dynamic payload length
	Reserved	7:6	0	R/W	Only '00' allowed
	DPL_P5	5	0	R/W	Enable dynamic payload length data pipe 5. (Requires EN_DPL and ENAA_P5)
	DPL_P4	4	0	R/W	Enable dynamic payload length data pipe 4. (Requires EN_DPL and ENAA_P4)
	DPL_P3	3	0	R/W	Enable dynamic payload length data pipe 3. (Requires EN_DPL and ENAA_P3)
	DPL_P2	2	0	R/W	Enable dynamic payload length data pipe 2. (Requires EN_DPL and ENAA_P2)
	DPL_P1	1	0	R/W	Enable dynamic payload length data pipe 1. (Requires EN_DPL and ENAA_P1)
	DPL_P0	0	0	R/W	Enable dynamic payload length data pipe 0. (Requires EN_DPL and ENAA_P0)
ID	FEATURE			R/W	Feature Register
	Reserved	7:3	0	R/W	Only '00000' allowed

	EN_DPL	2	0	R/W	Enables Dynamic Payload Length
	EN_ACK_PAY	1	0	R/W	Enables Payload with ACK
	EN_DYN_ACK	0	0	R/W	Enables the W_TX_PAYLOAD_NOACK command

Note: Don't write reserved registers and registers at other addresses in register bank 0

Register bank 1:

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
00		31:0	0	W	Must write with 0x414B01F2
01		31:0	0	W	Must write with 0xC04B0630
02		31:0	0	W	Must write with 0xA0FCC4000
03		31:0	0x03001200	W	Must write with 0x17003560
04		31:0	0	W	Must write with 0x4199000B (1Mbps data rate mode) or 0x4199100B (2 Mbps data rate mode) Write with 0x41110421 with make BK2411 transmit CW at the given channel
	XTALFC	16:13		W	Crystal offset compensation, center at 8. User can adjust this register to compensate crystal offset
05		31:0	0	W	Must write with 0x24017FBE
	RSSI_TH	29:26		W	RSSI Threshold for CD detect 0: -97 dBm, 2 dB/step, 15: -67 dBm
06		31:0	0	W	Must write with 0x00004000
07		31:0	0	W	Must write with 0x00000000
	RBANK	7		R	Register bank selection states. Switch register bank is done by SPI command "ACTIVATE" followed by 0x53 0: Register bank 0 1: Register bank 1
08	Chip ID	31:0	0	R	BEKEN Chip ID 0x61616161 (BK2411) 0x64646464 (BK2412)
09			0		Must write with 0x00000000
0A			0		Must write with 0xF6F54EF6
0B			0		Must write with 0xD651185C
0C		31:0	0		Please initialize with 0x2D005540
	TX_LOCK_SEL	10:9		R/W	Tx PLL lock time selection 0: 120 us, 1: 200 us, 2: 300 us, 3: 500 us
	TX_RAMP_SEL	8:5		R/W	Tx PA ramping time selection 0: 10 us, 1: 20us, 2: 30 us, ... , 15: 160 us
0D	NEW_FEATURE	31:0	0		Please initialize with 0x00007000 (1 Mbps data rate mode) or 0x00000400 (2 Mbps data rate mode)
	CYST_ACCU	14:12		R/W	Crystal accuracy, the worse accuracy requires the larger value Using 7 for crystal accuracy to about 100 PPM Using 0 for crystal accuracy better than 5 PPM

		11		R/W	Reserved
	MODU_MOD	10		R/W	Modulation type 1: FSK mode 0: GFSK mode
	GFSK_BT	9		R/W	1: BT = 0.5 0: BT = 1
	LONG_PL	8		R/W	Enable long payload in normal payload mode 1: Enable long payload mode 0: Normal payload mode
	LEN_LONG	7:0		R/W	Payload length for maximum 255 bytes payload
0E	RAMP	87:0	0x8fbde72aa 16523081 43081	W	Ramp curve Must write with 0xCFF7CF208104082081041
Note: Don't write reserved registers and no definition registers in register bank 1					

编程指南

配置 BK2411 模块是通过 SPI 方式进行的，可配置为 FIFO 方式和直接方式，我们推荐 BK2411 工作于 FIFO 收发模式，这种工作模式下，系统的程序编制会更加简单，并且稳定性也会更高，因此，下文着重介绍把 BK2411 配置为 FIFO 收发模式的器件配置方法。

使用BK2411模块无需掌握任何专业无线或高频方面的理论，读者只需要具备一定的C语言程序基础即可。其中配置BK2411主要包括载波频率、调制方式、数据发送速率、CRC校验、前导码、同步字、数据头、地址等，本文档没有涉及到的问题，读者可以参考BK2411官方手册或向我们寻求技术支持。同时，为便于用户开发，我们提供系列配套评估套件，为产品开发保驾护航，使无线应用开发大大加速，并避免不必要的误区。

程序分析

命令寄存器宏定义

//详细请参照英文 PDF 文档第 16-17 页

#define READ_REG	0x00	//读操作寄存器命令
#define WRITE_REG	0x20	//写操作寄存器命令
#define RD_RX_PLOAD	0x61	//读接收到数据命令
#define WR_TX_PLOAD	0xA0	//写待发送数据命令
#define FLUSH_TX	0xE1	//刷新发送 FIFO 命令
#define FLUSH_RX	0xE2	//刷新接收 FIFO 命令
#define REUSE_TX_PL	0xE3	//数据重发命令
#define W_TX_PAYLOAD_NOACK_CMD	0xb0	//在发送模式下使用，关闭自动应答
#define W_ACK_PAYLOAD_CMD	0xa8	

联系电话: 13704018223 陈工
在线咨询: QQ:35625400 474882985

E-mail: chj_006@sina.com
MSN: 1188mm88@hotmail.com

//在接收模式下使用，将数据和 ACK 一起写到通道 0 的 FIFO

```
#define ACTIVATE_CMD      0x50      //用于激活处于掉电模式或待机模式
#define R_RX_PL_WID_CMD   0x60      //从 FIFO 最高开始读数据
#define NOP                0xFF      // 空操作
```

工作方式配置寄存器宏定义

//详细请参照英文 PDF 文档第 19-24 页.

```
#define CONFIG              0x00
//配置寄存器。主要配置工作方式
#define EN_AA              0x01
//各通道自动应答设置寄存器，打开或关闭通道 P0-P5 的 ACK
#define EN_RXADDR          0x02
//各通道选择寄存器，用于打开或关闭通道 P0-P5
#define SETUP_AW           0x03
//地址长度寄存器，用于设置地址长度，最小 3 字节，最大 5 字节
#define SETUP_RETR         0x04
//设置自动发送时间间隔
#define RF_CH              0x05
//工作频率设置寄存器，用于设置无线工作频率
#define RF_SETUP           0x06
// RF 模式配置，用于设置功率大小，空中速率等
#define STATUS             0x07
//状态寄存器
#define OBSERVE_TX         0x08
//发送检测寄存器
#define CD                 0x09
//载波侦听寄存器，用于检测空中是否有无线信号
#define RX_ADDR_P0         0x0A      //      P0 接收地址
#define RX_ADDR_P1         0x0B      //      P0 接收地址
#define RX_ADDR_P2         0x0C      //      P1 接收地址
#define RX_ADDR_P3         0x0D      //      P2 接收地址
#define RX_ADDR_P4         0x0E      //      P2 接收地址
#define RX_ADDR_P5         0x0F      //      P3 接收地址
#define TX_ADDR            0x10      //      发送地址
#define RX_PW_P0           0x11      //      P0 接收数据长度
#define RX_PW_P1           0x12      //      P1 接收数据长度
#define RX_PW_P2           0x13      //      P2 接收数据长度
#define RX_PW_P3           0x14      //      P3 接收数据长度
#define RX_PW_P4           0x15      //      P4 接收数据长度
#define RX_PW_P5           0x16      //      P5 接收数据长度
#define FIFO_STATUS        0x17      //      FIFO 状态寄存器，检测 FIFO 状态
#define PAYLOAD_WIDTH      0x1f      //      address 发送数据长度设置寄存器
```

联系电话: 13704018223 陈 工
在线咨询: QQ:35625400 474882985

E-mail: chj_006@sina.com
MSN:1188mm88@hotmail.com

//RF2411 状态寄存器，详细请参照英文 PDF 文档第 19 页

```
#define STATUS_RX_DR      0x40    // 数据接收状态标志位
#define STATUS_TX_DS      0x20    // 数据发送状态标志位
#define STATUS_MAX_RT      0x10    //
#define STATUS_TX_FULL     0x01    //
```

状态寄存器宏定义

//详细请参照英文 PDF 文档第 23 页

```
#define FIFO_STATUS_TX_REUSE 0x40
#define FIFO_STATUS_TX_FULL  0x20
#define FIFO_STATUS_TX_EMPTY 0x10
#define FIFO_STATUS_RX_FULL  0x02
#define FIFO_STATUS_RX_EMPTY 0x01
```

自动应答寄存器宏定义

```
#define REG2_BUFFER_COUNT 0x02
#define REG3_STATUS       0x03
#define ASK_Rx_Full       (1<<7)
#define ASK_Rx_Empty      (1<<6)
#define ASK_Tx_Full       (1<<5)
#define ASK_Tx_Empty      (1<<4)
#define ASK__Reserved     (1<<3)
#define ASK__Irqn_Tx_Err   (1<<2)
#define ASK_Irqn_Rx_Done   (1<<1)
#define ASK_Irqn_Tx_Done   (1<<0)
#define ASK_Irqn_ALL (ASK__Irqn_Tx_Err|ASK_Irqn_Rx_Done|ASK_Irqn_Tx_Done)
```

BK2411 SPI 读写时序操作

```
char SPI_RW(char _char)
{
    char bit_ctr;
    for(bit_ctr=0;bit_ctr<8;bit_ctr++)    // output 8-bit
    {
        RF2411_MOSI = (_char & 0x80);    // output 'char', MSB to MOSI
        _char = (_char << 1);             // shift next bit into MSB..
        RF2411_SCK = 1;                   // Set SCK high..
        _char |= RF2411_MISO;              // capture current MISO bit
        RF2411_SCK = 0;                   // ..then set SCK low again
    }
}
```

```
    return(_char);                // return read char
}
```

BK2411 SPI 写寄存器操作

```
void SPI_Write_Reg(char reg, char value)
{
    RF2411_CSN = 0;                // CSN low, init SPI transaction
    op_status = SPI_RW(reg);        // select register
    SPI_RW(value);                  // ..and write value to it..
    RF2411_CSN = 1;                // CSN high again
}
```

BK2411 SPI 读寄存器操作

```
char SPI_Read_Reg(char reg)
{
    char value;
    RF2411_CSN = 0;                // CSN low, initialize SPI communication...
    op_status=SPI_RW(reg);          // Select register to read from..
    value = SPI_RW(0);              // ..then read registervalue
    RF2411_CSN = 1;                // CSN high, terminate SPI communication
    return(value);                  // return register value
}
```

工作模式初始化

```
unsigned char code RegArrFSK[][2]=
{
    {0, 0x0e},
    //CONFIG 配置, IRQ 引脚状态用于指示状态信息, 配置为 16 位 CRC 校验, 内部上电开启,
    {1, 0x3F},    //EN_AA ,所有通道都允许自动应答功能
    {2, 0x3F},    //EN_RXADDR, p0-p5 通道全部打开
    {3, 0x03},    //SETUP_AW, 地址长度设置为 5 字节
    {4, 0x31},    //SETUP_RETR
    {5, 0x17},    //RF_CH, 工作频率设置为 2.423GHZ
    {6, 0x15},    //RF_SETUP, {6, 0x17}, //1Mbps 速率, 功率设置为最大 5db
    {7, 0x07},    //STATUS ,配置为数据包传输模式
    {8, 0x00},    //OBSERVE_TX
    {9, 0x00},    //CD
    {12, 0xc3},   //RX_ADDR_P2
    {13, 0xc4},   //RX_ADDR_P3
}
```

```
        {14, 0xc5},      //RX_ADDR_P4
        {15, 0xc6},      //RX_ADDR_P5
        {17, 0x20},      //RX_PW_P0
        {18, 0x20},      //RX_PW_P1
        {19, 0x20},      //RX_PW_P2
        {20, 0x20},      //RX_PW_P3
        {21, 0x20},      //RX_PW_P4
        {22, 0x20},      //RX_PW_P5
        {23, 0x00},      //FIFO 状态寄存器
        {28, 0x3F},      //支持可变长数据包模式
        {29, 0x07}
};
```

BK2411 发送模式设置

```
void SwitchToTxMode(void)
{
    unsigned char value;
    SPI_Write_Reg(FLUSH_TX, 0);           //刷新 Tx-FIFO
    value=SPI_Read_Reg(CONFIG);           //读取 CONFIG 状态
    value=value&0xfd;                     //POWER DOWN 掉电
    SPI_Write_Reg(WRITE_REG + CONFIG, value);
    value=value&0xfe;
    SPI_Write_Reg(WRITE_REG + CONFIG, value); //接收模式
    value=value|0x02;
    SPI_Write_Reg(WRITE_REG + CONFIG, value); //POWER ON 上电启动，并切换
    为发送模式
}
```

BK2411 数据发送流程

```
void RF2411_SendPacket(unsigned char* buf, unsigned char len)
{
    unsigned char temp=0;
    SPI_Write_Buf(W_TX_PAYLOAD_NOACK_CMD, buf, len);
    RF2411_Delay(400);
    do
    {
        temp = SPI_Read_Reg(0x07);
    }while(!(temp&0x70));                //判断 TX_DS，是否发送完毕
    SPI_Write_Reg(0x07|WRITE_REG, 0x70); //刷新状态寄存器，为一次一收发数据准备
}
```

BK2411 接收模式设置

```
void SwitchToRxMode(void)
{
    unsigned char value;
    SPI_Write_Reg(FLUSH_RX, 0);           //刷新 Rx-FIFO
    value=SPI_Read_Reg(STATUS);           //读取工作状态
    SPI_Write_Reg(WRITE_REG+STATUS, value);
    // clear RX_DR or TX_DS or MAX_RT interrupt flag
    value=SPI_Read_Reg(CONFIG);           //读取 CONTIG 状态
    value=value&0xfd;
    SPI_Write_Reg(WRITE_REG + CONFIG, value); //先 POWER DOWN 掉电
    value=value|0x01;
    SPI_Write_Reg(WRITE_REG + CONFIG, value); //设置为发送模式
    value=value|0x02;
    SPI_Write_Reg(WRITE_REG + CONFIG, value); //上电启动, 并切换为接收模式
    RF2411_CE=1;
}
```

BK2411 接收数据流程

```
void RF2411_ReceivePacket(unsigned char* buf)
{
    unsigned char payloadLen;
    payloadLen=SPI_Read_Reg(R_RX_PL_WID_CMD);
    SPI_Read_Buf(RD_RX_PLOAD, buf, payloadLen);
}
```

无线应用注意事项

- (1) 无线模块的 VCC 电压范围为 1.8V-3.6V 之间, 不能在这个区间之外, 超过 3.6V 将会烧毁模块。推荐电压 3.3V 左右。
- (2) 除电源 VCC 和接地端, 其余脚都可以直接和普通的 51 单片机 I/O 口直接相连, 无需电平转换。当然对 3V 左右的单片机更加

适用了。

(3) 硬件上面没有 SPI 的单片机也可以控制本模块，用普通单片机 IO 口模拟 SPI 不需要单片机真正的串口介入，只需要普通的单片机 IO 口就可以了，当然用串口也可以了。模块按照接口提示和母板的逻辑地连接起来

(4) 标准 DIP 插针，如需要其他封装接口，或其他形式的接口，可联系我们定做。

(5) 任何单片机都可实现对无线模块的数据收发控制，并可根据我们提供的程序，然后结合自己擅长的单片机型号进行移植；

(6) 频道的间隔的说明：实际要想 2 个模块同时发射不相互干扰，**两者频道间隔应该至少相差 1MHZ**，这在组网时必须注意，否则同频比干扰。

(7) 实际用户可能会应用其他自己熟悉的单片机做为主控芯片，所以，建议大家在移植时注意以下 4 点：

A: 确保 IO 是输入输出方式，且必须设置成数字 IO；

B: 注意与使用的 IO 相关的寄存器设置，尤其是带外部中断、带 AD 功能的 IO，相关寄存器一定要设置好；

C: 调试时先写配置字，然后控制数据收发

D: 注意工作模式切换时间

我们的承诺

最后，欢迎您使用我们的产品，在应用中有技术问题请及时向我们联系，我们会予以技术知道，同时运输中出现产品问题我们会全面责任并予以更换。

愿与您一起走向成功